# Summary

The aim of this project is to provide Morrisons with a computerised specification management system. There have been significant problems with the current paper-based system with the main issue because it has not been feasible to conduct searches on the increasing backlog of paper specification forms. Morrisons would like a new system to store and modify their specifications in a manner where it is possible to conduct searches and retrieve the information they require. The motivation in undertaking this project is to help Morrisons achieve the possibility of performing searches on stored data and obtain the information that they need with greater convenience.

This project has helped Morrisons achieve the possibility of being able to search through specification forms for the information they require. The implemented database can be used to store, view and search Specification Forms and therefore all the minimum requirements have been met and exceeded. Additional functionality provided by the system gives the user on-line access to a user guide and be able to create ad-hoc reports on the information stored in the database.

# Acknowledgements

On completing this project, I would like to acknowledge the following people:

- Stuart Roberts my project supervisor for his invaluable help throughout this project.
- Duncan Roscoe from Morrisons for his input, feedback and for providing this challenging project.
- Alan Bromwich for his dedication, patience and proof reading.

# Contents

**SUMMARY** ..... **1**

**ACKNOWLEDGEMENTS** ..... **2**

**CHAPTER 1 - INTRODUCTION** ..... **5**

**1.1 INTRODUCTION TO MORRISONS** ..... 5
**1.2 STATEMENT OF THE PROBLEM** ..... 5

**CHAPTER 2 – BACKGROUND RESEARCH** ..... **6**

**2.1 INTRODUCTION** ..... 6
**2.2 METHODOLOGY** ..... 6

**CHAPTER 3 – PROJECT SCHEDULE** ..... **9**

**3.1 MILESTONES** ..... 10
**3.2 PROJECT PROGRESS** ..... 11

**CHAPTER 4 – ANALYSIS AND OBJECTIVES** ..... **12**

**4.1 INTRODUCTION** ..... 12
**4.2 ANALYSIS OF THE CURRENT SYSTEM** ..... 12
**4.3 USER REQUIREMENTS** ..... 13
**4.4 OBJECTIVES** ..... 14
**4.5 ANALYSIS OF INFORMATION TECHNOLOGY AT MORRISONS** ..... 15

**CHAPTER 5 - DESIGN** ..... **17**

**5.1 INTRODUCTION** ..... 17
**5.2 DATABASE DESIGN** ..... 17
**5.3 SPECIFICATION FORM DESIGN** ..... 26
**5.4 DESIGN OF DATA TRANSFER BETWEEN THE FORM AND DATABASE** ..... 26
**5.5 DESIGNING THE USER MANUAL** ..... 27

**CHAPTER 6 - IMPLEMENTATION** ..... **28**

**6.1 INTRODUCTION** ..... 28
**6.2 DATABASE IMPLEMENTATION** ..... 28
**6.3 IMPLEMENTING THE ELECTRONIC SPECIFICATION FORM** ..... 34
**6.4 DATA TRANSFER METHOD BETWEEN THE FORM AND DATABASE** ..... 36
**6.5 IMPLEMENTATION OF THE USER MANUAL** ..... 38

**CHAPTER 7 - TESTING** ..... **39**

**7.1 UNIT TESTING** ..... 39
**7.2 INTEGRATION TESTING** ..... 40
**7.3 USER ACCEPTANCE TESTING** ..... 40

**CHAPTER 8 - DEPLOYMENT** ..... **43**

**8.1 THE DATABASE** ..... 43
**8.2 THE SPECIFICATION FORM** ..... 43
**CHAPTER 9 – EVALUATION** ..... **44**
**9.1 INTRODUCTION** ..... 44
**9.2 DESIGN EVALUATION** ..... 44
**9.3 IMPLEMENTATION EVALUATION** ..... 46
**9.3 OVERALL SUMMARY** ..... 49
**BIBLIOGRAPHY** ..... **50**
**APPENDIX A** ..... **52**
**APPENDIX B MEETING DETAILS** ..... **53**
**APPENDIX C ER-DIAGRAM** ..... **54**
**APPENDIX D ER-MODELLING DOCUMENTATION** ..... **55**
**APPENDIX E DATABASE SCHEMA** ..... **57**
**APPENDIX E TESTING RESULTS** ..... **60**
**APPENDIX G USER MANUAL** ..... **64**

# Chapter 1 - Introduction

## 1.1 Introduction to Morrisons

Morrisons is the UK's fastest growing supermarket chain, being the fifth largest supermarket chain in the country with an annual turnover of £3.5 billion. To date they have 114 supermarkets around the country with 44,000 employees working for the company. The company sells more than 20,000 different products including well-known brands and their own Morrisons branded range of products. The Morrisons brand is quite extensive considering it accounts for 55% of the volume of products sold by the company.

## 1.2 Statement of the Problem

Currently Morrisons has around 5,000 to 6,000 of its own branded products sold throughout the supermarket. The decision to sell a new branded product is made by a team of buyers within Morrison's head office who fill in a set of paper specification forms related to the new product. With food laws becoming more restrictive, Morrisons decided that they need a system to store specification data so that they can conduct searches on the information and retrieve ad-hoc reports if needed. Presently with the paper-based process in place, it is not feasible to carry out a search on a range of job specifications. They approached the university to help them to automate the current manual process into a system that allows them to get information out of it. The focus of this project will therefore be to automate the specification process into a system that can be implemented and used hence meeting the user expectations.

# Chapter 2 – Background Research

## 2.1 Introduction

In order to understand the nature of the problem being solved, the background research undertaken has been focused on two main areas. The first area has involved looking into an appropriate project approach that can be used in attempting to solve the problem and the second area has involved reading about the various subject areas such as Human Computer Interaction (HCI) that are part of the Information Systems (IS) development process. To gain insight into how to approach an IS project, it has been necessary to read aspects of project management books [1], [2] helping towards the development of a suitable project management plan within the allocated time. Reading selective reports from last year's final year projects has helped to investigate the project approach taken by fellow students [3], [4]. One good IS development practice requires that a methodology be used as a framework for guidance within the project approach. This has been addressed by reading different IS development books that have helped in the decision to choose the most appropriate methodology for this particular project. This is discussed and evaluated in section 2.2 with all references included in the Bibliography. The other area of research has involved drawing upon material taught in previous School of Computing modules. Notes from module DB11 [5] proved to be informative for the ER – modelling exercise and the design of a relational database using normalisation. HCI issues were studied in order to implement a user-friendly system by taking into account good design practices [6]. This material also included useful areas on how to test and evaluate a system from an end user point of view that was taken into account at the end of the project. This project aims to address a specific problem faced by Morrisons. Currently there is no automated system already in place and subsequently all reading and investigations that have taken place have been on the grounds of developing a new system to meet user requirements.

## 2.2 Methodology

### 2.2.2 Background

A methodology is defined as "a collection of procedures, techniques, tools, and documentation which will help the systems developer in their efforts to implement a new information system."[7]. Reading through the numerous IS books it is appreciated that methodologies differ greatly in their techniques and emphasis on tackling systems development problems and some are more suitable for tackling this project than others. The following list of requirements should be encouraged by the methodology chosen for this project, as they are seen as guidelines that can help towards producing a successful system [8].

- To get an accurate requirement of the user needs
- To promote a systematic method of development where progress can be monitored
- To allow for any changes in the development process
- To provide a solution within a time scale
- To produce a well documented system that is easy to maintain
- To provide a system that is accepted and liked by those affected by it.

### 2.2.3 Methodology evaluation

The traditional Systems Development Life Cycle Model is a methodology that involves working through the various stages of a project from determining user requirements to the implementation and testing of a system [9]. Its advantage is that it is a tried and tested method and refers to five of the six objectives listed in the previous section. Its output driven approach leads to inflexibility in the design process, if changes are needed and this could in turn lead to outright user rejection of the system at the end. The use of this method involves an element of risk if user requirements were to change at any time during the process. The Structured Systems Analysis and Design Methodology (SSADM) attempts to build-upon the traditional life cycle and address some of its problems. However, the problem of the lack of iteration built into the life cycle remains, making the methodology too rigid.

The Rapid Application Development (RAD) methodology has been developed to speed up the development process. RAD is a variant of the Information Engineering methodology (IE) that covers all aspects of the life cycle, though RAD is identified, as a methodology in it's own right. RAD involves building successive prototypes of the system, incorporating changes gathered from user feedback until a system meeting the final requirements has been developed [10]. This approach is considered suitable for the project as it takes into account the short amount of time available to produce the final system and focuses on the issue of user uncertainty that could occur throughout the project by following an iterative process. The choice of RAD as a methodology is based on how well the framework incorporates each of the requirements in section 2.2.1 but most importantly, it addresses the elements of inflexibility and the time that is considered the largest constraints for this project. Using RAD means a quick prototype resembling the end system can be shown to users during the implementation stage for their feedback, which in turn can be incorporated back into the final system. The use of prototyping also allows components of the system to be developed independently of the others without reliance on the whole system to work as a whole and thus focus on the task in hand [11].

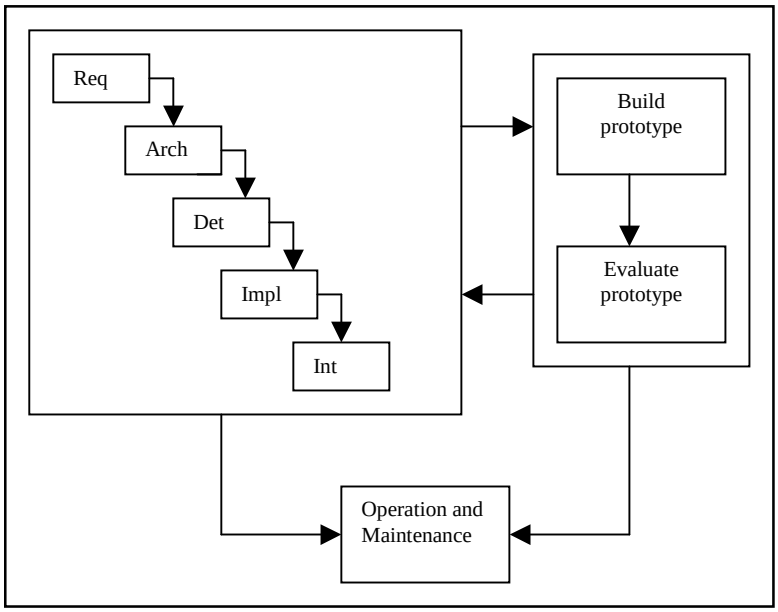



**Fig.1 Evolutionary prototyping throughout the life cycle**

Evolutionary prototyping has been chosen as the prototyping approach as it allows the prototype to be built upon and evolve over time into the final system [12]. By encouraging user participation and incorporating user feedback modifications can be made where necessary thereby improving the final system and decreasing the chances of user rejection at the end. Another issue being tackled by prototyping is that it is not always possible to get the entire user requirements at the beginning of the cycle as user needs can change over time and prototyping can account for this problem. I intend to adapt and tailor the needs of the RAD methodology to the requirements of this project by incorporating the traditional life cycle model into the methodology for the analysis and design sections hence providing a more robust framework to work by than if using RAD alone.

I will be using the 'MoSCow rules' [13] detailed in the analysis section, for deriving, categorising and prioritising the user needs. The 'MoSCow rules' are part of the theory of Dynamic Systems Development Methodology which is a framework designed for developing particular types of systems when using the RAD approach. The DSDM framework other than the 'MoSCow rules' will not be used for this project. Having decided on a methodology it is now possible to provide the following outline of a project approach for this project to follow:

- Identify the problem
- Derive the user requirements
- Identify the possible approaches with a feasibility study
- Undertake systems analysis
- Undertake systems implementation
- Testing
- Evaluation of the system

# Chapter 3 – Project Schedule

A decision made about the project approach allows a project schedule to be drawn up below (Fig.2) setting out clear timescales to help achieve the major milestones set out. The milestones show the main stages identified in order to reach a solution to the problem with details of the tasks involved at each stage. A detailed Gantt chart in the project progress section shows an outline of the tasks set out within the timescales and the progress made at each stage with slippages from schedule clearly illustrated.

| |
|---|
| **Background Research (2 weeks)** |
| This section of the project discusses the background reading carried out around the project, evaluating the various methodologies that can be used for IS projects and the method chosen. Furthermore, an outline of the project approach that will be followed to carry out this project is discussed. |
| **Analysis and Objectives (5-6 weeks)** Deliverable: Aim & Min. Requirements Form 2nd Nov 01 |
| The analysis section involves studying the current situation so that reasons can be determined about why an automated system is needed. Based on this analysis, user requirements have been derived with a list of objectives. In addition, a study is carried out on the technology suitable for implementing the system in this project along with a list of the constraints to be considered. |
| **Design (4 weeks)** Deliverable: Mid Project Report 13th Dec 01 |
| The design section is split into several parts discussing the design, problems faced and how the user requirements have been met for each component of the system. The main sections focus on the specification form, the database and the data transfer tool linking them that allows the system to function as a whole. |
| **Implementation (12 weeks)** Deliverable: Draft Chapter and Table of Contents 18th Mar 02 |
| In this section, the implementation of the final solution is examined categorised by each main component. This involves showing how the database was constructed, how the specification forms were produced and the tasks involved in connecting the system together. |
| **Testing (2 weeks)** |
| The testing section looks at the aim of testing, how the test plan was derived and the results from unit, integration and user acceptance testing. |
| **Deployment (1 week)** |
| This section discusses the work involved to install the system at Morrison's head office. |
| **Evaluation (1 week)** Deliverable: Final Year Report 1st May 02 |
| A final evaluation is given in this last part of the report along with reflections and suggestions for the project in the future detailed in Appendix A. |

**Fig. 2 Project Schedule table**

## 3.1 Milestones

The project period spans over a period of seven months and therefore milestones identifying critical tasks within project phases have been extracted from the Gantt chart. This helps to keep to the project plan and check that progress is at the stage it should be. Slippages in time are documented on the Gantt chart along with an adjustment of tasks. The list of key tasks identified as important to producing the end solution is:

- **Database Design**

  This stage involves designing the database relations for data to be stored. Without this database structure, it is not possible to see how data on the specification forms will be saved or how functionality required by the end user will be carried out if there is no data model to test entities upon. ER Modelling and Normalisation are part of the database design process.

- **Specification Forms Implementation**

  The forms must be implemented before any work on the mapping between the database and the forms can begin. This is so that the logical order of how data from the specification forms is transferred to the database can be solved.

- **Mapping between the database and forms**

  Completion of this stage will link the database and forms into the final system. Once it has been realised how the data will be read into the database the process needs to be implemented so that any problems that arise can be solved.

- **User Evaluation and Testing**

  This stage will involve Morrisons testing and evaluating the system to ensure that their needs have been met. Unit and integration testing will be carried out before user acceptance testing.

## 3.2 Project Progress

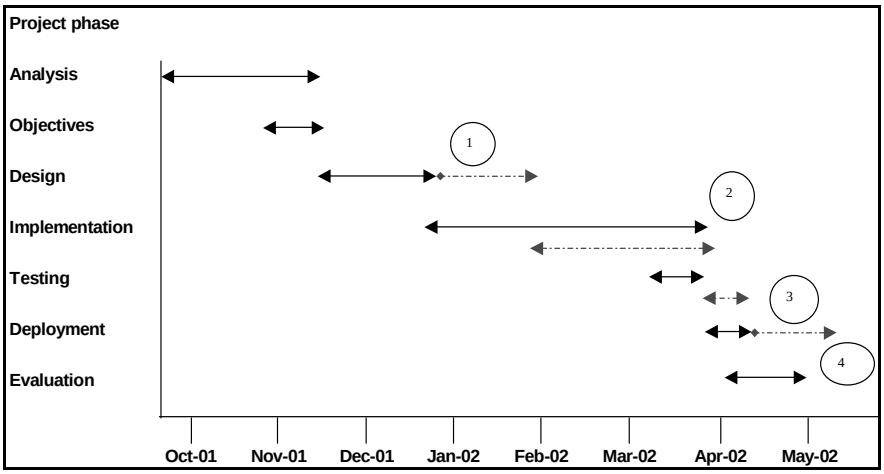



**Fig. 3 Gantt chart**

The Gantt chart shows the sections of the project schedule mapped against the project timeline. The black arrows indicate the time scheduled to be spent on each of the tasks. The dashed arrows labelled with corresponding numbers show areas of project slippage. These are briefly discussed below against the corresponding number on the Gantt chart with exact details referenced throughout the project.

1) The first slippage in the project was that of the design section. Work on this stage was on schedule until Morrisons produced a new format of specification forms just as the work was coming to a completion at the beginning of January, ready for their inspection. This delayed the design work by four weeks as major adjustments had to be made to the design of the electronic form and the database schema.

2) Consequently, this had a knock-on effect and the tasks scheduled after the design section were delayed with the time for the implementation reduced by four weeks. This put pressure for the prototype to be completed as fast as possible with full functionality. The trade off for doing this was that for the user feedback and testing tasks, the graphic design work was not finished, but all functionality had been implemented.

3) The testing including user acceptance testing took place during the time scheduled for deployment of the system. As a result, deployment has been delayed until after the completion of the project report an arrangement agreed with both parties. The evaluation of the system has been completed on time as scheduled.

# Chapter 4 – Analysis and Objectives

## 4.1 Introduction

To gain an understanding of the requirements for an automated specification system it is necessary to carry out an analysis of the current system from the user perspective. Thus, a meeting was arranged with a long-term user from Morrisons to discuss the present system in place, plus any additional suggestions for an automated system. Existing fixed documents such as a specification form was used as a guideline for the type of documentation Morrisons used as part of the current process. The information gathered in the following section is taken from a meeting held with Stuart Roberts, the project supervisor and Duncan Roscoe from Morrisons. Full details of the meeting are provided in Appendix B.

## 4.2 Analysis of the Current System

The current specification management system in place is paper-based. A team of buyers make decisions on whether to sell a branded product. If team approval is given, a unique job number is assigned to the product on a specification form with some other details about the product also filled in. This form is then forwarded onto a pre-determined supplier who is required to fill in the rest and once complete return it to Morrisons via email by scanning it or mail. If Morrisons approve the contents of the specification form, it is signed and filed away in a drawing cabinet. If the form is rejected it is returned back to the supplier to repeat the process until Morrisons give their approval or the product specification is completely abandoned. The following is a list of advantages and disadvantages of the current system provided from an end user viewpoint:

**Advantages of the current system**

- Storing specification forms in filing cabinets is cheap.
- A procedure that all staff involved is familiar with; a new process would require training with the new system.
- A paper-based file system means there is a hard copy of all specification forms filled in.

**Disadvantages of the current system**

- Paper-based process is a time consuming process with forms prone to being mislaid.
- Storage space is becoming a problem.
- Suppliers do not always understand information that is required of them and therefore forms often have to be returned in order to be filled out as required and the process is unnecessarily delayed further.

- Suppliers often change food legislation information pre-filled by Morrisons on the specification form even though it is clearly stated that it should not be altered.
- Search for information involves a manual process that is often infeasible.
- There is a duplication of work when generating reports needed by other departments such as marketing
- No way of tracking deleted products therefore redundant information exists in the paper based system.
- Human error from difficulty in interpreting handwriting on forms

## 4.3 User Requirements

From the details in the previous section, it is now possible to derive the user requirements of the end user. An analogy used in RAD projects is the MoSCoW rules [14], which help to determine the user requirements. The four parts of this concept helps to divide the varying users requirements based on their relative importance to one another and usage in the final implementation of the system. Due to the short amount of time to implement a system of this scale, this analogy helps to prioritise what is implemented in the final solution. This categorisation of the requirements has been agreed with the end user and is documented in the details of the meeting held in Appendix B.

The first part is the **'Must have'** requirements that "are fundamental to the system"[15]. These have been identified as:

- A specification form designed for data entry that can be completed and submitted electronically to Morrisons.
- A fully implemented database that holds the specification forms
- A user manual to guide operation of the system

The second part is the **'Should have'** requirements. These are important requirements, but the system would still operate without them. They are identified as:

- The data in the database must not be accessed by unauthorised personnel and be password protected.
- Implemented reporting tools to create ad-hoc reports
- Security measure that does not allow parts of the specification form already filled into be altered.
- Flags to highlight certain critical changes in the system

The third part is the **'Could have'** requirements, which would have benefit to the system, but could be left out of the development process:

- Highlight old data that is past a certain date for deletion
- Provision of concurrent access to the system

The fourth category is **'Want to have but will not have this time around'**.

- To combine the current suppliers contact database system with the proposed new system to solve duplication of effort on entering contact details.

## 4.4 Objectives

From the preceding analysis of the current system and the user requirements, it is possible to derive a set of objectives for the project:

1. **Automate the data entry process from a set of paper forms to an electronic document.** Suppliers must be able to receive, fill-in and return electronic forms. The paper-based specification forms will remain as an option for non-technical suppliers.
2. **Design and implement a database to hold and view the data from specification forms.** Part of implementing a database means to solve the data transfer problem. This entails providing a way to enter data from the forms into the system automatically and should not require duplication of effort by using a direct interface to the database for data entry.
3. **Ensure that the system is scalable to cope with an increasing amount of data and data held is maintainable.** The database should be able to cope with the increasing number of specification forms generated.
4. **Provide a detailed user manual describing the operations of the database.** All software implemented should be documented providing any new users requiring training in the future and for version control of the database.

### 4.4.1 Minimum Requirements

The minimum requirements set for achieving the objectives are:

1. An analysis of the user requirements resulting in a database designed to hold specification data sent from suppliers.
2. An electronic specification form document designed for data entry.
3. A user manual to accompany the system once deployed.

### 4.4.2 Further enhancements

Some enhancements identified that could greatly improve system usability are:

1. Investigate other technologies that can be used for data entry other than an electronic document.
2. Design a comprehensive search tool that can generate reports based on filtered information.
3. Provide users with different levels of security for system access.
4. Allow saving half completed data in the system that can be re-written to. Any changes made to the original data would need highlighting.
5. Provide on-line guidance

## 4.5 Analysis of Information Technology at Morrisons

### 4.5.1 Hardware and Software Analysis

The Morrisons Head Office where the automated system is needed has one PC with external network access. This PC will be used for the sending and receiving of the proposed electronic specification forms. All other PC's used by administration staff are part of an internal intranet but do not have access to external networks. This is due to the high level of security within Morrisons, who fear from an outside attack on their internal systems. The software on all PC's currently consists of Access 97, which holds a supplier contact database and a standard Microsoft Office Suite used for administrative tasks. The networked PC also has the above software with the addition of Internet Explorer 5, used for browsing the Internet.

### 4.5.2 Project Constraints

The meeting with Duncan Roscoe from Morrisons revealed alongside user requirements the constraints involved in undertaking this project.

- **No remote access to Morrisons network**

  It has been specified that any system implemented should not be available for remote access by external clients or any other Morrisons' offices over the external network. Suppliers as a result will not be granted access via an interface to enter specification form data directly into any database, as this will involve granting network access and is unacceptable to Morrisons. The specification system for that reason will need to be client based where access is limited to a few members of staff. A text attachment in email is considered acceptable to the networks of Morrisons and its suppliers, but other executables are rejected if there is the possibility it may transfer viruses into the internal network.

- **Willing to invest in software if needed.**

The company are willing to invest in software needed for the new system but have given a strong preference for using their existing database software (Access 97) as they already have other databases using this software that they may like to integrate with the new system in the future. Additionally the Morrisons staff is familiar with navigation around this software and would therefore only require minimum training using the new system itself.

- **Preference of software for Electronic Forms**

A preference has been stated for using their existing software Word 2000 for designing the electronic specification forms as the majority of their suppliers have Microsoft Word installed and any other software would require purchases of new software on their behalf resulting in a large financial cost. It is outside of the scope of this project to check on the accuracy of this statement but the assumption made is that all Morrisons' suppliers will have Internet access for sending and receiving electronic forms.

# Chapter 5 - Design

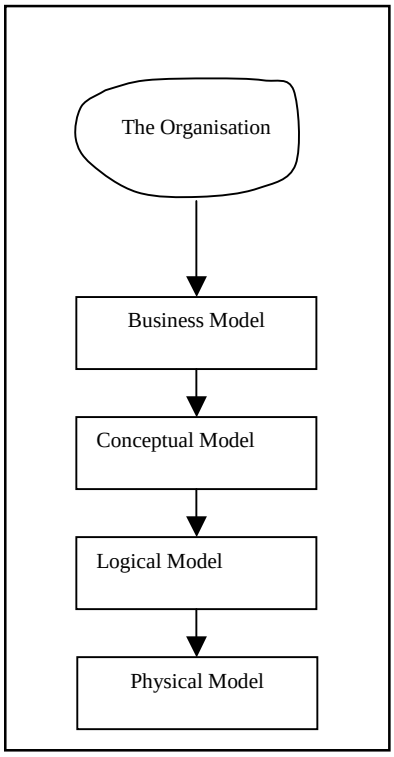



**Fig. 4 Approach in outline**

## 5.1 Introduction

This chapter is divided into two sections that focus on the main aspects of the design of the system and show how the implementation stage in the next chapter was reached. The analysis carried out in Chapter 4 and derived list of user requirements provided the background needed to continue with the design process. The database approach used throughout the project has been that of a traditionally used model in database development (Fig.4)[16]. Section one discusses the different elements of database design such as the data analysis techniques used to derive ER models and normalised relations known as the conceptual model. The logical and physical models illustrated by Fig.4 are part of the implementation stage and will be discussed in the subsequent chapter. Section two discusses the considerations taken into account for designing the electronic specification forms.

## 5.2 Database Design

### 5.2.1 Data Modelling

The first requirement in database design is deciding on the information that needs to be stored in the database. The approach to data analysis combined of using a document and interview driven technique to ensure everything that needed to be represented in the database was included. This is often known as the ‘universe of discourse’, which is the area that has been defined for analysis. In this case, the paper specification forms represented the area for analysis. By referring to the specification forms it is possible to think about constructing an Entity-Relationship model also known as the conceptual model using the Chen notation (1976). Chen’s ER model uses entities and relationships between entities as representation of the information in the real world [17]. An entity is defined as “something we may want to hold data about”, an attribute as “descriptive values of entities” and relationships as something that “represents an association between two entities” [18].

A top-down approach to ER-modelling was used where entities were identified first and then relationships between them were defined. This model was built-upon by filling in more details of

attributes and defining key attributes of entities [19]. Key attributes uniquely identify any entity occurrence. The key entities and associated attributes derived using the definitions above are shown in Fig.5.

| | |
|---|---|
| **SUPPLIER** | 'Supplier_number', 'supplier_name', 'address', 'Tel' |
| **FACTORY** | 'Factory_name', 'address', 'Tel' |
| **SPECIFICATION CONTACT** | 'Emerg_name', 'Job_title', 'Tel' |
| **24H EMERGENCY CONTACT** | 'Spec_name', 'Job_title', 'Tel', 'Email' |
| **JOB** | 'Job_no', 'supplier_number', 'Factory_name', 'product_name', 'food name', 'supplier comments', 'protective_atmosphere', 'life of product once open', 'health mark', 'shelf life', 'shelf life format', 'sequence of packaging', 'serving instructions', 'primary_trans_pack_inlimit', 'material reuse', 'materials_recoverable', 'noxious_sub_ltd', 'mats_comply','plastic_food_comply', 'product group', 'trading mgr', 'issue date', 'return date', 'approved date', 'initials', 'HACCP study', 'shelf trials', 'weight', 'drained weight', 'volume', 'count', 'not applicable', 'e mark', 'qty approx', 'irradiated', 'GM', 'suitable for veg', 'doc procedure for package', 'spec complete date', 'RDA field', 'NutInfo', 'PerServing', 'claims_info', 'Ingredient', 'Quid', 'warnings', 'analysis', 'frequency', 'reason', 'Mspec', 'Allergens', 'ingredient formulation', 'ingredient description', 'country of origin', 'percentage', 'category', 'qty per grams', 'qty per servings', 'energy kjperg', energy kcapperg' |

**Fig. 5 Entities and their Attributes**

The next phase involves defining relationships between the entities, the most common being one-to-many relationships, for example the diagram below shows that the entity 'SPECIFICATION CONTACT *takes on many* JOBS' but a 'JOB *only allocates one* SPECIFICATION CONTACT'.

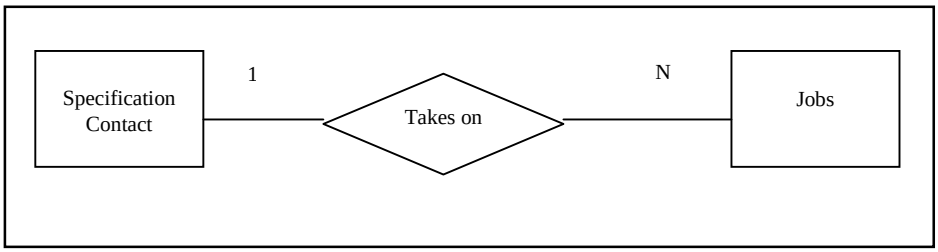



**Fig. 6 Entity Relationship Diagram**

Several refinements have been made to the data model before an entity-relationship diagram correctly representing the processes can be finalised. For example the Supplier and Factory was a many-to-many relationship where a 'SUPPLIER *supplies many* FACTORIES' and a 'FACTORY *supplies many* SUPPLIERS'. In this situation it is better to split these relations into two one-to-many relationships with a third intermediate entity linking the entities together [20] as illustrated in Fig.7.

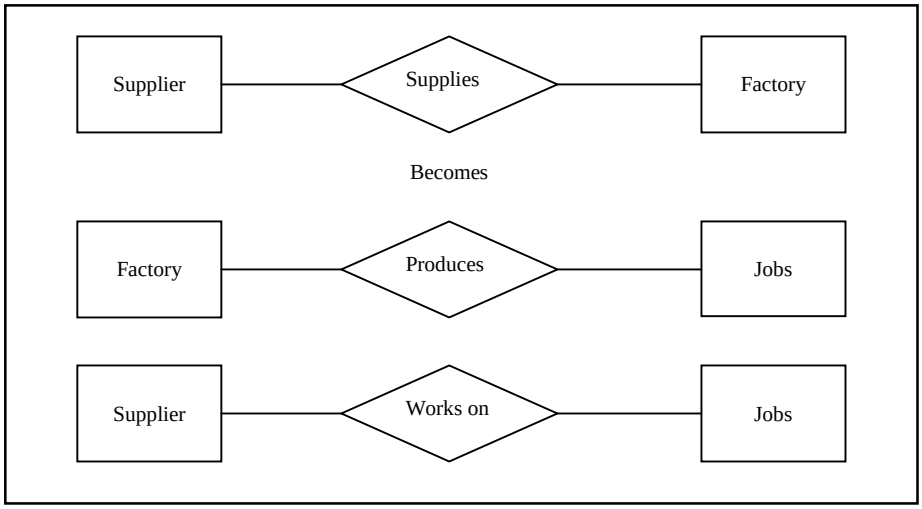



**Fig.7 Entity-Relationship Modelling**

The new entity JOB '*may be only supplied by one* SUPPLIER' and '*is only produced by one* FACTORY. Thus a 'SUPPLIER *supplies many* JOBS' and a 'FACTORY *produces many* JOBS' but a 'JOB *only has one* SUPPLIER' and '*is produced by only one* FACTORY'. This process is known as the flattening of relations and helps to build a flexible data model more adaptable to any changes in the future.

A meeting with Duncan Roscoe using an entity-relationship diagram has helped to consolidate the fact that all the entities and correct definition of relationships between them have been defined. The diagram proved to be a useful tool for Duncan's understanding and provided the feedback to ensure the accuracy of the model. Accuracy is a crucial aspect of the design process and could result in significant development problems later on if not correct at this stage. The significance of user involvement at this stage is that it allows the user to feel they are participating towards the system and helps to create a partial acceptance of it as it continues to develop. Appendix C shows an Entity-Relationship diagram representing all the entities and relationships in full.

### 5.2.2 Normalisation

As the entities have been defined as relations, it is possible to apply rules based on primary keys and functional dependencies to the relational model that will simplify the relations by minimising data redundancy and anomalies. This process known as normalisation attempts to provide a flexible data model through the method of extensive data analysis [21].

To reach Third Normal Form (3NF) where all non-key attributes must be independent of one another, first each entity must be complete. This is the definition of First Normal Form (1NF) and ensures that there is only one possible value for each domain and not a set of values. The other condition to reach 3NF is that all attributes of an entity must be functionally dependent on the entire key commonly

known as Second Normal Form (2NF). If this is not the case then the attributes functionally dependent on only part of the key should be split off into a separate relation with the key they are fully dependent on [22].

The Supplier, Factory, 24H Emergency Contact and Specification Contact relations shown in Fig.8 are all in 2NF already as all the attributes of the relations are dependent on their respective keys, that is the value of A uniquely determines the value of B. They are also in 3NF as no non-key attributes are dependent on other non-key attributes known as 'transitive dependencies'.

| **A** | → | **B** |
|---|---|---|
| Supplier_number | → | Supplier_Name, Address, Tel |
| Factory_name | → | Address, Tel |
| Emerg Name | → | Job_title, Tel |
| Spec Name | → | Job_title, Tel, Email |

**Fig. 8 Functional Dependency Representation**

However, the same cannot be said of the job relation where many attributes have been identified as being dependent on non-key values as well as the primary key (Job_no) and therefore are not in 2NF. As noted, these attributes should be split into separate relations and this has resulted in the following extra entities in the relational model:

PRODUCT ALLERGENS, PRODUCT MICROBIOLOGICAL SPECIFICATIONS, PRODUCT NUTRITION, PRODUCT COMPOSITION, WARNINGS, PRODUCT INGREDIENT, PRODUCT CLAIMS

These new entities and the job relation have been checked to ensure they meet 3NF. Appendix D includes detailed documentation about all the entities, relationships between them, attributes and key attributes recorded in a data dictionary.

### 5.2.3 Database Schema

The schema below shows some of the table structure designs for a relational database. The entities in Appendix D map directly to the tables and this should not change unless the user requirements or application requirements change in some way. Key attributes become primary keys. A full database schema listing all the tables is included in Appendix E.

**PRODUCT FORMULATION**

| ATTRIBUTE | ATTRIBUTE TYPE | DESCRIPTION |
|---|---|---|
| Job_no | Text | Foreign key for the job table |
| Ingredient formulation | Text | Ingredient name |
| Ingredient description | Text | Description of ingredient |
| Country of origin | Text | Describes origin of ingredient |
| Percentage | Text | Percentage of ingredients |

**FACTORY**

| ATTRIBUTE | ATTRIBUTE TYPE | DESCRIPTION |
|---|---|---|
| Factory_name | Text | Primary Key for Factory table |
| Address | Text | Address of Factory |
| Tel | Text | Telephone of Factory |

### 5.2.4 System Functionality

The final stage of the data analysis is to ensure that the normalised data model meets the system requirements outlined in Chapter 4 hence meeting the objectives set out for the system. These requirements however, have to be defined into detailed events with a corresponding list of operations that is needed to carry them out. This is very difficult and any errors could result in changes required during the development process that could be costly in terms of time and financially [23].

A way of determining the functionality of the system is to model some of the most frequent transactions that will be carried out by users. Although it is not possible to cover every condition, the informal ‘80-20’ rule states “80% of the workload is represented by 20% of the most frequently used transactions, which governs design” [24]. The next paragraphs note some of the most common transactions envisaged for the system where the data model has been tested to ensure it can support the flow of operations needed to carry out a task.

**1) Browsing situation**

1. Open the database
2. A welcome screen with a list of options will greet users
3. Select ‘Search for a job specification’ where user encounters the Job form
4. User selects a Job number to preview all the information associated with it
5. User scrolls through the information until the end and either
6. Prints a report of the selected specification
7. OR
8. Exits the application, returning to the welcome screen.

**2) Loading specification form into database situation**

1. Open the database
2. A welcome screen with a list of options will greet users
3. Select ‘Load a specification form ’ option where user encounters the Job form showing information of the loaded specification form
4. User scrolls through the forms viewing the information, making modifications where necessary
5. User reaches the end of the forms and either
6. Prints a report of the specification form viewed
7. OR
8. Exits the application returning to the welcome screen

**3) Supplier modifications situation**

1. Open the database
2. A welcome screen with a list of options will greet users
3. Select ‘Edit supplier option’ where user encounters the supplier form
4. User selects a supplier to edit/delete and any changes are saved on closure of the form
5. User exits the application returning to the welcome screen

**4) Factory modification situation**

1. Open the database
2. A welcome screen with a list of options will greet users
3. Select ‘Edit factory option’ where user encounters the factory form
4. User selects a factory to edit/delete and any changes are saved on closure of the form
5. User exits the application returning to the welcome screen

The situations above have been turned into a list of functions required for minimum usability of the system.

**1) Job function**

1. Load a specification form in
2. List job details by selecting a job number
3. Edit job details
4. Delete a job

**2) Factory**

1. List factory details by selecting a Factory name
2. Edit factory details
3. Delete a factory

**3) Supplier**

1. List supplier details by selecting a Supplier name
2. Edit supplier details
3. Delete a supplier

### 5.2.5 Integrity Constraints

The relational model has two constraint rules known as the entity integrity constraint and the referential integrity constraint. The entity constraint rules states that no primary key value should be null. The referential integrity constraint is based on maintaining consistency amongst two relations and states that the database must not contain any unmatched foreign key values. An unmatched foreign key is a foreign key value for which there is no matching value of the relevant primary key in the relevant target relation [25]. Most Relational Database Management Systems (RDBMS) support integrity constraints as part of the data definition and rules do not have to be explicitly implemented but are recognised by the RDBMS when primary keys are specified and relationships between relations defined.

### 5.2.6 Query design

It is important to recognise the most commonly required data that needs to be viewed in the database so that queries can be implemented to achieve this. The production of a list can ensure that all the queries are included in the implementation and not missed out. One of the “should have” requirements involves the implementation of reporting tools and the following list will help to ensure that important details are not missed out when implementing the queries that produce a report.

- Show the name of the food, claims info, product quantity details, e-mark, approximate quantity, protective atmosphere details, shelf life once opened details, warnings, shelf life format, health mark, nutrition, ingredients, allergens and cooking instruction for every job.
- Show all the details for every job specified
- Show all the suppliers and their details
- Show all the factories and their details

### 5.2.7 Form design

In order to design accurate forms the following considerations describing the purpose of the forms is taken into account before any design work has begun.

1. The forms need to replicate the format of information that was previously held on paper, therefore the ordering and flow of information is an important design aspect when considering the end users.
2. The forms should only reflect the current data in the database and do not need to be designed for data entry but only the editing of information.

To satisfy the point "1" above it is necessary to document the order of information on the paper specification forms into sections that can be implemented as a set of forms in the database. This paper order will then be used to represent the flow of information between the database forms once implemented. According to the good design principles set out for HCI [26] end users should not notice the difference between reading information once represented on paper to that represented by forms. A paper specification form consists of seven pages and each page of data has a sub-heading relating to the data on that page. It thus seemed appropriate to use the sub-headings as titles for each of the database forms shown in Fig.9 keeping the relevant data under each heading where possible.

1. Job details
2. Claims Information
3. Ingredients
4. Dietary Information
5. Preparation and Cooking
6. Formulation
7. Packaging

**Fig.9 Flow of Information on Form**

A problem that could be foreseen was that there was too much information stored on each page of the paper forms. Attempting to map one page to one single form on-screen would result in cluttered information that would be more of an eyesore to the user than an advantage. The headings in the diagram were as a result divided into smaller sub headings to eliminate this problem. The first form of any application should consist of a menu screen that allows the user access to the functions of the system. This 'welcome screen' with a main menu will be designed to open automatically upon entering the application. The main options available to the user from the welcome screen will be:

1. Load a specification form in
2. Search for a job specification
3. Edit/Delete Supplier information
4. Edit/Delete Factory information

A description of each of the forms designed is below: (the forms highlighted in bold indicate the main forms from where the non-bold forms will be opened e.g. **Form 1**)

**Form 1** – This form is the first page of the paper specification form containing the main details of the job such as the job number. On this form, the user can open other forms that lead to the supplier, factory, 24-hour emergency and specification contact details relating to the job. On closure of this form, the user returns to form 1.

Form 2 – The Supplier Form, which can be edited.

Form 3 – The Factory details form.

Form 4 – The 24-hour emergency contact details form.

Form 5 – The Specification contact detail form.

**Form 6** – This form shows more data relating to the job, which users can edit if they wish. It will be implemented for all forms that any changes made to data will be saved as the user moves onto another form.

**Form 7** – This form shows Claims information relating to the job, again the same edit features will be available as the other forms.

**Form 8** – This form shows nutrition information for the job. Here the RDA form can be opened which returns the focus to form 8 when closed.

Form 9 – This form the RDA information.

**Form 10** – This form shows the Ingredient of the job.

**Form 11** – All allergen information is represented on this page and the user can open a form that shows control of composition of information. When this form is closed, form 11 is returned to focus.

Form 12 – This form shows the control of composition information.

**Form 13** – This form shows the cooking method related to the job and a form for viewing formulation information can be opened from it that returns focus to form 13 on closure.

Form 14 – This form shows the formulation information.

**Form 15** – This form shows the packaging information for a job.

**Form 16** – This form is designed for access from the main menu for editing or deleting Supplier contact details.

**Form 17** – This form is designed for access again from the main menu but this time for editing/deleting Factory contact details.

**Form 18** – This is the welcome screen form.

## 5.3 Specification Form Design

The design of the electronic specification forms is very important, as it will in essence be an indirect interface to the database. The consideration of HCI principles involved designing the electronic forms to match the format of the paper specification form. The logic behind this is the same applied to that designing the forms as part of the database design in that the ordering of tasks on the electronic form must be logical to the user and matches their same expectations of tasks when they fill in a set of paper forms. As the electronic forms are an exact match to the format and information on the paper specification forms, the user does not encounter any syntax, semantics or jargon that is unfamiliar to them and this helps to achieve a good natural interface [27]. The result of all this is a seven page electronic document resembling the paper specification form page for page with flow of information between the electronic documents following the order depicted in Fig.9.

### 5.3.1 Validation rules

Validation rules specify the data entry format that certain fields should adhere to. This is normally part of the database design that when implemented provides error messages to the user via the database interface. The justification for including the validation rules in this section is that the project approach involves using the electronic document for data entry and not the forms documented under section 5.2.7 (Form Design). Duncan Roscoe from Morrisons has checked the rules below for accuracy.

- The Supplier Field must only consist of only 4 integers.
- The date fields cannot be greater than today's date
- Nutrition fields must all be to 1 decimal place.
- Quid fields must only be integer values between 1 and 100
- Percentage field can only contain integer values between 1 and 100

## 5.4 Design of Data Transfer between the Form and Database

One of the minimum requirements in designing a database is to solve the data transfer problem between the database and the electronic specification form. The import feature in Access 2000 provides the functionality to transfer text into the database in numerous formats. This is an ad-hoc feature and there are no strict guidelines on how to design an import as there are for example designing a database. The design of this component of the system leading up to the implementation therefore involved systematically exploring the various ways the import feature could be used to solve the data transfer problem. A more comprehensive description of the exact method used is to follow in the Implementation chapter.

## 5.5 Designing the User Manual

The purpose of providing a user manual is to help users obtain the full value and benefits from the system. This is especially important for new users in the future who have not received any training. Several HCI and more specifically “Writing User Manual Guides” books have been consulted [28] in deciding on a design format for the manual. As a result, a vertical approach has been chosen for contents design where the manual illustrates how to do specific tasks and is organised according to the tasks carried out by a user. This therefore tells the user how to use the system with step-by-step instructions rather than merely describing it. The format of the report has been designed according to the following structure:

1. Front Page
2. Contents Page
3. How to load a specification form into the database
4. How to search for a job
5. How to edit/delete a supplier
6. How to edit/delete a factory
7. How to create a marketing report
8. How to create a specification report

The tasks included in the manual are those that the system is capable of carrying which includes all the minimum functionality plus ‘should have’ features required by the end user from the Analysis Section. This simplified approach has been used to produce a minimal manual because it only contains the essentials therefore focusing on the user tasks and emphasising error recovery. Experiments with this type of manual has shown that users learned to use the system 40% faster than when given a fuller manual [29]. The exact contents of the manual is not described here but will be discussed in the Implementation chapter where the user manual is referenced in the Appendices.

# Chapter 6 - Implementation

## 6.1 Introduction

This chapter examines the implementation stages of the various components that make up the whole system with four main sections discussing the work carried out for each part and the issues faced. Section one looks at the main stages in producing the database as the finer detail is outside the scope of this report. The next three sections subsequently describe the implementation of the electronic specification forms, the work involved to solve the data transfer problem between Microsoft Word and Access and the User Manual. All minimum requirements along with the 'must have' and 'should have' requirements have been fulfilled for the system.

## 6.2 Database Implementation

The mapping of the conceptual model to a form required by the relational database management system (RDBMS) is called the logical model and the RDBMS being used is Access 2000. The project constraints require that the Access 97 version software be used but once the database is implemented, it will be saved to a format that allows it to be compatible with Access 97. This is because the Access 97 software version is not available on the PC's in the School of Computing. To convert the conceptual model to the relational model the entities become relations, attributes map directly, the key attributes become primary keys of the relation and relationships become relations themselves [30].

### 6.2.1 Creating tables and relationships

The first task in implementing a database is to create the relations (tables) for data to be stored. The 'Table Design view' feature in Access helps to create tables where attribute names and their associated data-types, description and various other controls such as field size, validation rules and allowance of null values are set. Validation rules for reasons explained in the design section 5.3.1 have not been implemented as part of the database but will be applied as part of the electronic forms.

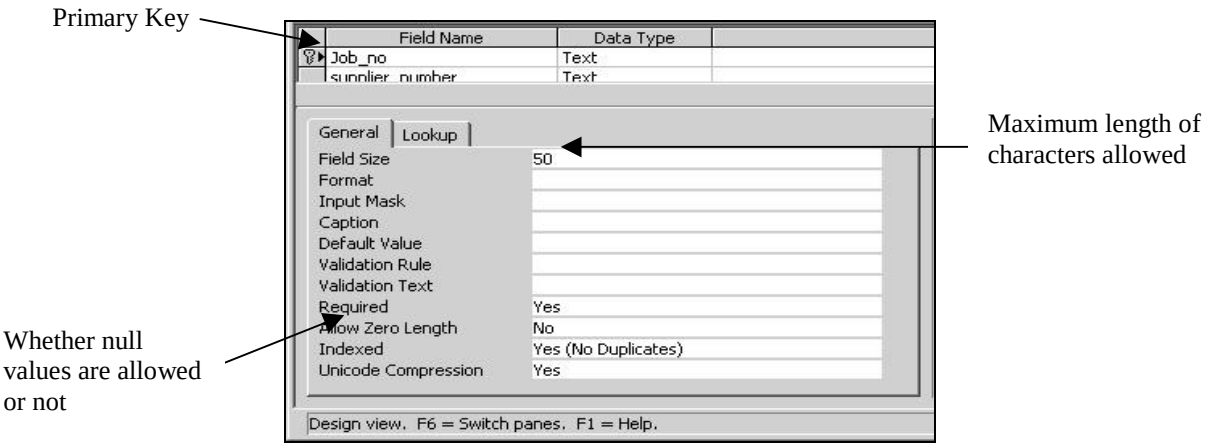



**Fig.10 Creating Tables**

The primary key is defined during table creation ready for the later task of defining relationships between the tables. To enforce referential integrity in the database no primary key created should accept a null value though Microsoft Access sets this as default. The next step involves creating relationships between the tables based on the relations defined in the ER-diagram. The screen shot below shows the 'Edit Relationship Tool' being used to create a relationship on the same attribute names in different relations by selecting the attributes from each table.

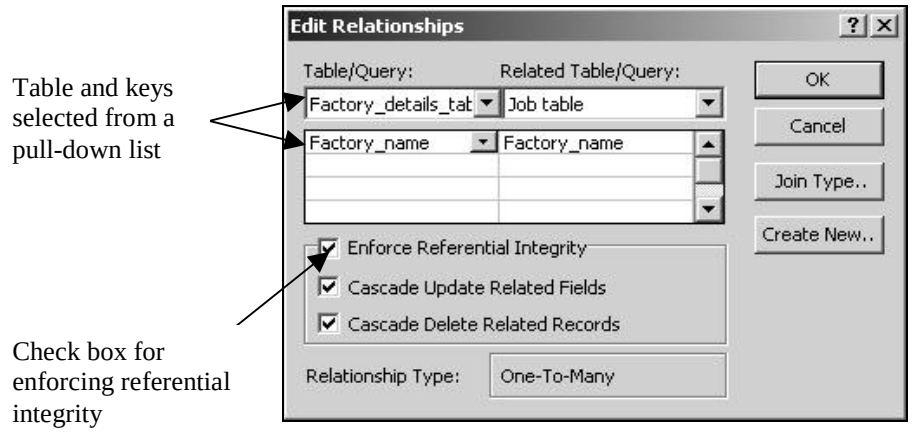



**Fig.11 Creating Relationships**

The check box for selecting 'Enforce Referential Integrity' is checked along with the 'Cascade Update Related Fields' and 'Cascade Deleted Related Records' options. Setting cascade options means that when records are deleted or the primary key value changes in a primary table, Access 2000 makes the necessary changes to related tables in order to preserve referential integrity. This means that data remains in a consistent state and relationships are not broken. If these additional cascade options are not set with the 'Enforce Referential Integrity' option selected then the rules of referential integrity would not allow deletes or updates to take place. When creating relationships Access 2000 also automatically creates a default join property that only includes rows where the joined fields from both tables are equal. This default has been kept for all tables [31].

### 6.2.2 Form Implementation

Implementing the forms was quite challenging with extra code manipulation required to get the forms to behave in the way intended. An Access 2000 form wizard was used to create and rename the forms based on the design set out in Chapter 5 with text box controls for representing the data and corresponding labels to describe the data held in text boxes. Opening the forms in design view allowed the remainder of the design implementation work to be finished along with the coding behind command buttons that would provide the functionality required on each form. The code behind command buttons was a mixture of running macros and Visual Basic (VB) code for times when greater code manipulation was needed. An Access macro consists of a list of actions that are carried

out, step-by-step, in response to an event for example when a form closes. Ideally VB should be used in favour over macros because if the form were exported to another database the functionality of the form would remain working as the VB code would be also be exported in contrast to when using macros. The following layout was a basis for implementing all forms except the welcome screen, on which extra controls were added to forms as needed. In total eighteen forms have been created.

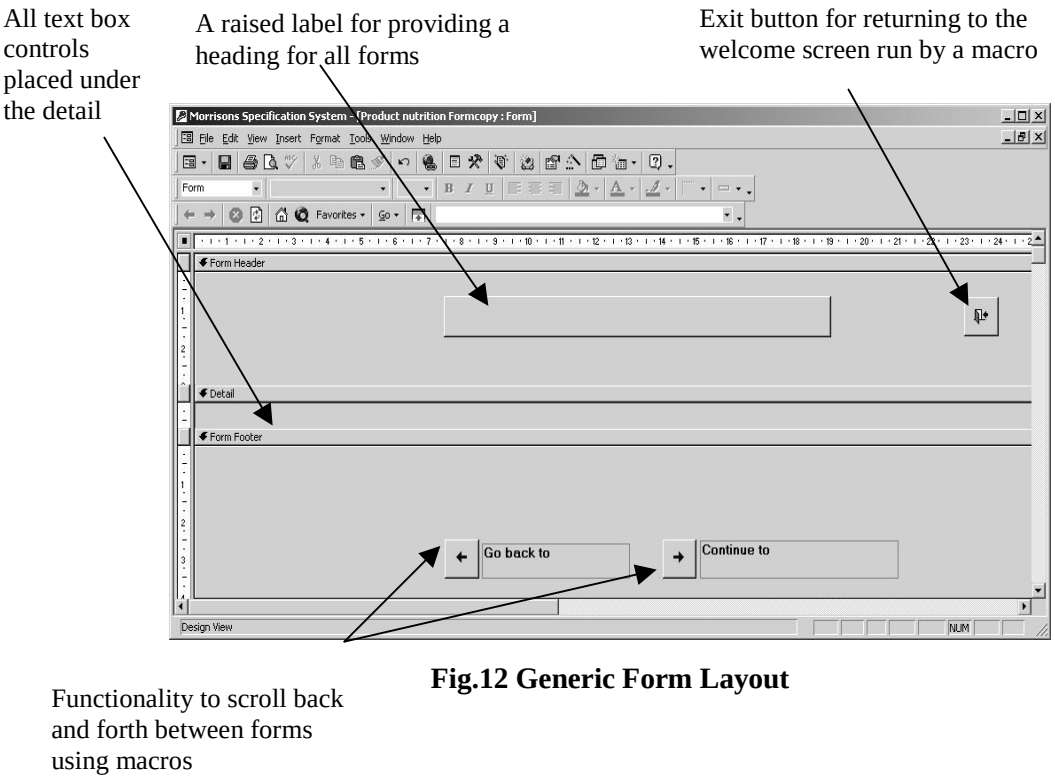

All text box controls placed under the detail
A raised label for providing a heading for all forms
Exit button for returning to the welcome screen run by a macro
Morrisons Specification System - [Product nutrition Formcopy : Form]
Form Header
Detail
Form Footer
Go back to
Continue to
Design View
Functionality to scroll back and forth between forms using macros

**Fig.12 Generic Form Layout**

The main form users encounter when opening the application is the 'Welcome Screen' implemented using a feature called Switchboard Manager. A switchboard is the user's primary point of entry into the application and it displays a list of actions that the user can carry out to work with the information in the database. The control panel shown underneath is set up for providing the various options designed for the main menu with functionality to carry out the actions selected from the options available. To choose the option of running a macro, first a macro must be designed and implemented to carry out the required action so that it can be found in a list of all the macros in the database. Macros have been used behind each of the command options available on the welcome screen form.

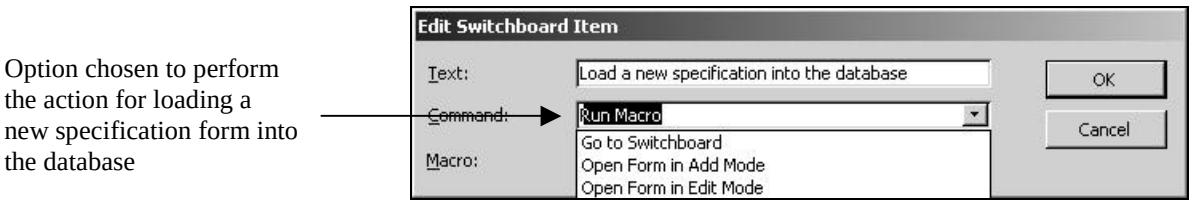



**Fig.13 Switchboard Manager Control Panel**

One of the objectives from the Analysis section in producing a database is to provide a database where specification data can be viewed. This has completed by the implementation of seventeen Access forms that represent all the information from one set of paper specification forms. The reason why there are so many forms is so that the information on each form does not appear cluttered to the end user, but is clear for their task. These forms are sequentially linked using macros behind the command buttons for moving between the forms. Links between Forms one to eight are via one another with access to the other seven forms provided from links within these forms. Standard features implemented on all forms includes opening the forms in edit mode, maximising the form to the full screen size on opening and on closure saving the form to save any changes that may have been made. The following paragraphs will discuss some of the unique features used to customise some of the main forms once created by the wizard.

**i) JOB FORM**

One of the key features implemented on this form is a Combo Box that allows the user to search for a job by selecting the job number from a list of job numbers in a pull-down list. The job numbers in this list are provided from a query implemented in the properties section of the Combo Box that selects all the job numbers from the 'job_no' field in the job table. Code implemented behind the Combo Box in the After_Update event (Fig.14) as part of the Class Module for the form is for when a user selects a job number to view data by and so that the rest of the controls on the job form show related information to the job number chosen. A Class Module is a collection of procedures and functions that exist in association with the forms created.

```
Private Sub Job_no_search_AfterUpdate()

' Find the record that matches the control.
Dim rs As Object

Set rs = Me.Recordset.Clone
rs.FindFirst "[Job_no] = '" & Me![Job_no_search] & "'"
If Not rs.EOF Then Me.Bookmark = rs.Bookmark
End Sub
```

**Fig.14 After_Update Event**

As information from the paper specification form is lay out over eighteen access forms the code above would not apply to the next form in the sequence. This means the data on the next form in the sequence would no longer be related to the job number chosen on the first form. To overcome this a macro included in the design for every form uses code whereby any form opened as part of the sequence has to display data based on matching values in the job number text box control. The Job_no is the primary key that relates the majority of relations. For design purposes, this Job_no text box is kept as a hidden control and only visible on Form 1 in order not to confuse the user by displaying it on every form but so that the condition can still be tested whenever a form is opened. The conditional

code **[Job_no]=[Forms]![Job Specification Form2]![Job_no]** is implemented behind command buttons on the form that allows the user to move back and forth between forms viewing related data from the same specification record. The form name is changed accordingly for each form in the sequence. A delete button allows the user to select a job using the Combo Box and delete it from the database.

**ii) SUPPLIER FORM**

The implementation of this form fulfils the requirement of being able to edit and delete supplier information. The justification for why only edit and delete functions are provided is that Morrisons already have a Contacts database with Supplier and Factory details and will therefore never manually add a new supplier or factory into the database. A Combo Box with customised code allows the user to select a supplier name where other controls on the form show the related information for the supplier chosen. When the user selects a supplier name from the Combo Box, the delete button erases the supplier from the database. The properties for the delete button includes a refresh command on the click event of the button so that when the user looks in the pull-down list of the Combo Box the deleted supplier is no longer in the list. Without the refresh, the form would have to be closed and re-opened for the changes to have taken place. To edit data, the user selects the required supplier name, makes the changes and presses the save command. The user then has the option of returning to the 'Welcome Screen'.

**iii) FACTORY FORM**

The implementation of this form is based exactly on the supplier form above. The user is able to select a factory name to edit or delete using a Combo Box based on the same principles above.

**iv) POP-UP FORMS**

Seven of the eighteen forms implemented have been designed so that they open from within other forms. These forms however, have been adapted to pop-up forms, which remain on top of other windows, as creating full-size forms are too large for the small amount of information being shown on them. A form is made to pop-up by selecting the pop-up option in the forms properties section when the form is in design view. The modal property of all pop-up forms has been set to 'YES' as this allows the forms to keep focus until they are closed.

### 6.2.3 Error trapping

Implementing forms means error trapping has to be performed. This is converting Access 2000 cryptic error messages into messages that the user can understand and that are informative enough to allow them to correct the error and continue with their task. Although the purpose of the forms is not for data entry editing of information is allowed and there may be occasions when the user edits a

primary key value so that it is left blank (known as a null value) or tries to enter the same key value in twice which violates integrity constraint rules. This results in the occurrence of a system error with a message generated that an end user will not always understand. It was necessary to generate these errors on the forms so that the system error code could be captured and then put into VB code, which intercepts the error when it next occurs. The code below is specified in the Class Module of the relevant forms under the Form Error event and is triggered whenever a system error matching the value specified in the code occurs. A user-friendly message instead of the system error is generated on error that allows the user to understand the problem and the code then returns focus to the control where the error has occurred so that the user can correct the problem.

```
Private Sub Form_Error(DataErr As Integer, Response As Integer)
Dim stMsg3 As String
'Error code 3022 for duplicate values'
Const conDupKey = 3022

If DataErr = conDupKey Then
   stMsg3 = "You have entered a duplicate Job Number. "
   stMsg3 = stMsg3 & "Please enter a unique number."
   MsgBox stMsg3
   Job_no.SetFocus
   Responses = acDataErrContinue

   End If
   End Sub
```

**Fig.15 Code to convert system errors**

### 6.2.4 Implementing reports

A 'should have' requirement in the Analysis section is to provide the user with a reporting tool that they can use to generate reports on records in the database. The report will be pre-formatted with the required data fields for the report provided in advance by Morrisons. This is to be known as the Marketing Report. Using the fields in the query design from section 5.2.6, the query below (Fig.16) has been implemented where the user clicks on a command button 'Print a Marketing Report' on the welcome screen.

```
SELECT DISTINCTROW [Job table].[Job_no], [Job table].[product_name], [Product claims
info].[claims_info], [Job table].[not applicable], [Job table].[weight], [Job table].[volume], [Job
table].[count], [Job table].[e mark], [Job table].[qty approx], [Job table].[protective_atmosphere],
[Job table].[life of product once open], [Warnings table].[warnings], [Job table].[shelf life
format], [Job table].[health mark], [Job table].[serving instructions]
FROM ((((([Job table] INNER JOIN [Product allergens table] ON [Job table].[Job_no]=[Product
allergens table].[Job_no]) INNER JOIN [Product claims info] ON [Job table].[Job_no]=[Product
claims info].[Job_no]) INNER JOIN [Product Ingriedient table] ON [Job
table].[Job_no]=[Product Ingriedient table].[Job_no]) INNER JOIN [Product nutrition table] ON
[Job table].[Job_no]=[Product nutrition table].[Job_no]) INNER JOIN [Warnings table] ON [Job
table].[Job_no]=[Warnings table].[Job_no]
WHERE ((([Job table].[Job_no])=[Enter job number for marketing report]))
ORDER BY [Job table].[Job_no];
```

**Fig.16 SQL Query for producing a report**

A pop-up screen appears asking the user to enter the job number that they would like to print a report on. Once a job-number is entered and confirmed, a report is generated that the user can proceed to print out. To create the reporting tool a query was written in Structured Query Language (SQL) using a query wizard in design view. Based on this, a report wizard was used to create a report founded on the query as the control source of information. In design view, changes have been made to the report to make the information appear in a logical order familiar to the user.

## 6.3 Implementing the Electronic Specification Form

The electronic forms for data entry are being implemented using Microsoft Word 2000. Extensive research has been done into the functionality of Word 2000 to ensure it can support the tasks that are required. Creating the functionality on the forms has involved using various features designed for building forms from a Form Toolbar.

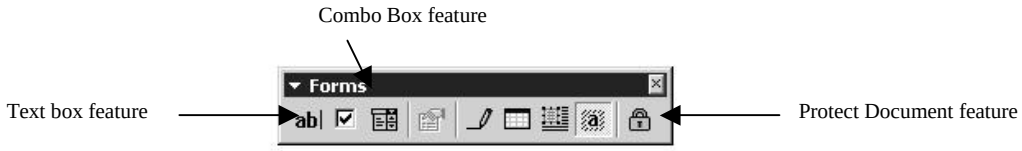



**Fig.17 Forms Toolbar**

The toolbar (Fig.17) has a Text Box feature that has been used to allow data entry by the user with a properties section where options can be set for each Text Box. The properties box has been used extensively throughout the implementation of each control used as its features allow it to be converted to a pull-down list for data entry that has a limited response set such as YES/NO values. In addition, data validation and format has been implemented at this stage for each of the Text Boxes thereby restricting values and adhering to validation rules required by the end user.

Another ‘should have’ requirements from the Analysis section is that data that has already been filled in on certain parts of the form should not be altered. This has been implemented using the toolbar feature protect document that grants the user of a form a password with which to protect changes once the form has been filled in. The users will be shown how to protect certain areas of the form by creating a password and how to regain access to the form and make changes. The areas filled in and protected by a password cannot be altered by anyone who does not have access to the password. The remainder of the form not protected by a password can be completed as normal.

Tables using the insert table tool have been implemented to separate the areas of information on the form with each table row fixed in size. This is done by right clicking on the table row that brings up the table properties for a row. Here the exact row height is specified so that the row cannot grow in size as the user types more data in a section. This is particularly useful for memo sections of the form

where the user could enter more data than is actually required of them. The reasoning behind fixed sized data-entry boxes is so that the format of the forms is not lost as the user enters information on the form otherwise rows would expand over to the next page of the document and confuse the user. Initially the forms had been produced using expanding rows but it was changed to a fixed size due to the lost formatting of the form. Consideration because of the fixed row size has to be given as to how much information can be filled in on one area. The final decision has been based on samples of filled in paper specification forms with a different length of information in certain sections and conversations with Duncan Roscoe from Morrisons about word limits. The screenshot below shows part of the form design and implemented features.

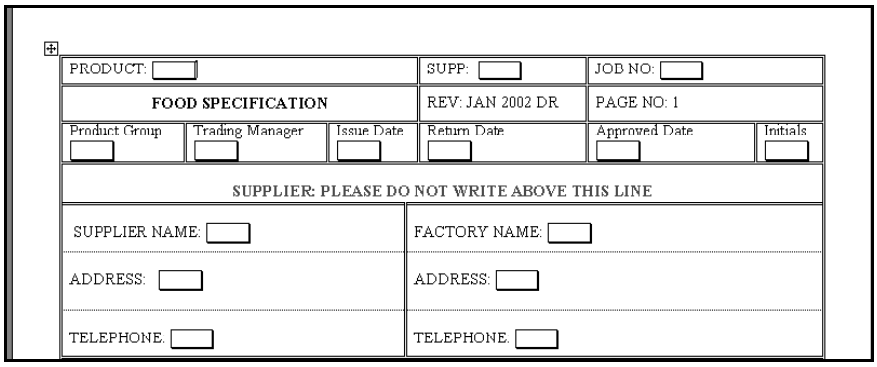



**Fig.18 Electronic Form format**

A demonstration of the implemented form was given to Duncan Roscoe for his feedback. The suggestions below were made to be incorporated into the final form. As a result, all of the points have been taken into consideration, as only minor changes were needed.

- Make the default value for the data fields of the format 00-00-00.
- Place a 'click to select' instruction next to all YES/NO drop down fields on the form
- Change the order of the fields in the Product Quantity section so that Not Applicable is the first field encountered
- Change all the Nutrition Values so that they have a default of 0.0 except energy values that should be zero.
- Expand the RDA field by eight to ten lines
- Put a % sign next to all 'Quid' labels on the form
- Change the default values of all packaging question to the order specified by Duncan
- Reduce the packaging method field by ten lines
- Change the default values of the allergens to the order specified by Duncan

## 6.4 Data Transfer Method between the Form and Database

Part of the minimum requirements in implementing a database is also to solve the data transfer problem by getting data from the electronic word document into the database in one action. This involves not having to manually re-enter all the information from Word to Access forms as this is a duplication of effort and is not feasible. The features of Microsoft Word and Access have been explored extensively to provide a suitable solution, as it is this component that binds the system together as a whole. The implementation of the electronic Word document allows data entry in fields to be saved as data for forms. This feature means text entered in data entry text boxes is saved as a comma delimited file where each data field is separated by commas and enclosed in speech marks.

The import feature of Access can be used to read in this text-only file into a temporary table created at the time of import. A problem encountered during implementation that could not be foreseen during the design is that Access only supports 256 fields in a table and the text file saved from the electronic document contained over 500 fields. A workaround involves using VB code that runs from a function called by a macro to split the text file into four smaller files overall. The code to achieve this called split_function was provided by Stuart Roberts and was modified to run from a Global Module in Access. The split_function code is instructed to recognise the data strings in a text file by recognising that every data string is enclosed by a pair of speech marks characters and separated by a comma character. When the function is called, two numbers are specified in the parameter to instruct the function which fields in the file it should count from and up to, in order to split it into another file. The parameters set out for splitting the large text file was strategically done so that data belonging to one section on the form was kept together where possible. The function has been thoroughly tested before being implemented to ensure that the text files have not been modified in any way or missing data as a result of the file split. The database has been implemented in a directory where the user will also be required to save the text file from a filled in specification form. This directory path has been specified in the split_function macro so that the function knows where to search for the text file. If the text file were not saved in the directory required, an error message would occur saying ‘file not found in C:\rest_of_path’.

The next stage is to automate the import procedure for each of the small text files by using macros. This has been manually tested to ensure that the import wizard used does not encounter any problems with each of the four text files. Import specification properties specified for each file set options for how the text file should be read into the database and the character to recognise for separating the fields in the file, which in this case is a comma character. The manual import procedure revealed no errors and therefore a macro were implemented to automate the four separate imports. A problem faced with the actual import tool itself is that the file being read in can only be saved to one table.

This is an issue because the text files being imported contain data that originates from fields in many tables in the database. To overcome this problem it has been necessary to create temporary tables at the time of import to hold the data in. This presents an extra difficulty of how to remove the data from these tables into the correct relational tables.

Four temporary tables holding data are created because of the import and the data in these tables need to be removed and appended to the relational tables they are intended for. This problem has been attempted to overcome by writing SQL queries that access each field in the temporary tables and append the data from the fields to the correct tables. Writing all the queries needed has been a time consuming process that was not scheduled for as part of the implementation phase resulting in project slippage. In hindsight, a more suitable approach with reference to time in the future would be to automate this process by writing VB code to loop through the fields in the temporary tables and retrieve the information to be appended. This should make the database easier to adapt if any changes are required unlike the approach that has been used.

An element of error checking has been included in all queries so that if a primary key is already in the database the append function does not take place. Normally, if an attempt was made to add a duplicate key to the database, Access would generate an error but this should be avoided during the automatic load of a specification form, as we do not want the user to experience any errors during this process. A macro has been used to run all the queries in a logical order that will not violate any referential integrity rules and insert the data into the tables. It has been tested extensively to ensure that all data appended is in the correct tables and correct any problems that occur from running the queries collectively under one macro.

All the objects created as part of the load process must be deleted such as the temporary tables and text files as a new instance of these will be created every time the load macro runs. If these were not deleted, the user would encounter file duplication errors in the load process and this should be avoided at all costs. The deletion of file objects was achieved by writing VB code that accessed objects from the File System Object Library. The DeleteFile object deletes objects that are outside of the database from a function called within the database. A macro has been used to call the function that has been coded in a Global Module to carry out a delete operation on the text files created because of the import. Another action inside the macro deletes the temporary tables created from the import.

The final task in solving the data transfer problem is to create a macro that will collectively run all the other macros created in a pre-determined logical order. This large macro provides the functionality required to load a specification form into the database in one action and runs behind a command

button so that the user is unaware of the actions occurring behind the scene. This is the ‘Load a Job Specification in’ button on the welcome screen.

## 6.5 Implementation of the User Manual

The user manual accompanying the system has been produced in Microsoft Word 2000. The design format of the manual outlined in section 5.5 was used to create each section of the guide. Under each section that describes the tasks, corresponding screenshots from the system have been included for further guidance. Each heading on the contents page is a hyperlink which when clicked on refers the user to that section in the manual hence the user does not have to scroll through the guide looking for information. The full user manual can be found in Appendix G. As a further enhancement to the requirements, the user manual is part of an on-line guide accessed from within the database instead of paper document. This has been achieved with VB code that opens the User Manual Document in Word. Extra command buttons have been implemented on all of the Access 2000 database forms that run this code when clicked on so that the manual can be accessed from anywhere within the system. The manual is also available on the welcome screen as an option to the user. The reasoning behind providing access to the manual from anywhere within the system is for availability and consistency factors concerning HCI principles. The advantages of on-line documentation over a paper manual are that paper manuals can get lost and tend to be physically confined to one place [32].

# Chapter 7 - Testing

Testing software is an important aspect of the software design life cycle after the implementation stage has been carried out. The reasons for testing are that to ensure that the system works as it is intended and that it is accepted by the end users and will be used in the future. Several Software Engineering books [33], [34] have been consulted to help construct a detailed and relevant test plan for this project. The types of testing carried out have been unit testing, integration testing and user acceptance testing, equal of importance. Unit testing involves testing every component of the system to look for the existence of errors. The purpose of this is to correct errors ensuring that they do not occur once the system is deployed and cause it to fail. Integration testing verifies that the components of a system work together as described in the design specifications ensuring the system is ready for the final stage of acceptance testing. User Acceptance testing involves participation of the users in operating the system to check that all requirements have been met. Unit and Integration testing is known as program-level view of testing whereas user acceptance testing is known as a system-level view of testing where we check that the system solves the requirements defined during analysis and is not rejected by the user. The summary of results for each type of testing is shown below with test plans and a full set of results for all tests in Appendix F.

## 7.1 Unit Testing

Unit testing involves testing every part of the software for bugs, validating the correctness of data input and checking integrity rules are being adhered in order to guarantee the performance and robustness of the final system. Many of the components were thoroughly tested during the implementation so that minimal problems would have to be dealt with during the final testing. To ensure that the tests done are thorough, a logical approach has been used to verify whether all the parts of the system are running, as they should. The test requires trying to stop the system working by doing tasks that a user would do on a day-to-day basis. The test plan is based on looking at every aspect of the forms in the database by selecting values in Combo Boxes, editing data as well as testing features on the electronic form.

### 7.1.1 Summary of results

Overall, the unit testing was quite successful as a lot of testing was carried out through the development in order to minimise problems at the end though there are a few areas of the system that could be improved. When a job number, which does not exist in the database is entered for a report a blank report is generated. An enhancement for the future would be to provide an error message if an invalid job number is entered so that a blank report is not generated. Another weakness revealed is

that the user does not get any feedback after a delete or edit successful operation has been carried out. This is justified as the main users are not novice users and a message box confirming the success of an operation on every occasion would more likely be a hindrance than a benefit.

## 7.2 Integration Testing

The objective of user acceptance testing is to discover any errors in the interface once the components of the system are linked together. There is only one component being used to link the database and electronic form together and it should not be difficult to determine the cause of an error should one occur. The testing entails insuring that the desired output is achieved from the system after a pre-determined input. The test plan is based on the output on all forms in the database after a text file has been inserted into the system.

### 7.2.1 Summary of results

The test plan for integration testing required testing the load function by using the electronic form as an input and seeing if the desired output was achieved. On first attempt, the load was unsuccessful as one of the queries reported a null violation and therefore the load was abandoned. Having searched for and fixed the problem the second attempt was successful though the load takes about 10 seconds to complete.

## 7.3 User Acceptance Testing

User Acceptance testing is part of the assessment for guaranteeing whether a system meets the user needs and is thus ‘accepted by the user’. To see if these requirements have been met, the system can be evaluated against the objectives set out for it at the beginning. This is discussed further in the evaluation section. Another way of assessing user needs is to observe the user interacting with the system. This can be done in a number of ways but the RAD methodology requires user involvement throughout the implementation as the chances of acceptance are greater if the user has been part of the development process. Every opportunity has been taken to provide an active role for the user during the process thereby decreasing the chance of outright rejection at the end.

The final acceptance testing process has consisted of a set of pre-determined functionality tasks given to the user to carry out on the system. This has been in the form a pilot test [35] where the system has been installed on an experimental basis and the tests rely on the ‘everyday working’ of the system. The user during the course of action has been observed and encouraged to think aloud by criticising the system or mentioning the pleasing aspects so that all comments can be noted. In reality this type of testing uncovers more than differences with the user requirements but allows the user to check what is really wanted whether it has been specified in the requirements or not. This is a comparison of the end

product to the current user needs. Duncan Roscoe was available to carry out user acceptance testing on behalf of Morrisons head office. The results of the testing are shown below in Table 1.

### 7.3.1 Summary of results

The user seemed satisfied with the system though did not find some of the forms very intuitive. This was put down to the user being unfamiliar with the system and would not be the case once more experience was gained. The user was very impressed by the load feature once it could be seen that it had been successful by viewing the data on forms. Some extra features were mentioned throughout testing such as being able to save a job record from the database back into the original electronic specification form. This has been noted as an extra feature that can be implemented in the future but overall, the objectives set out in the Analysis section have been achieved.

| TEST | FEEDBACK |
|---|---|
| Filling in a specification form? | Impressed by the resemblance of the form to the paper form and how form is fixed in size so does not expand when you are writing |
| Where invalid data format is input do error messages show up? | Yes, errors show up and require re-entry. However the help tips in the corner of the screen were not immediately noticed |
| Is the specification form easy to navigate around and easy to tab through? | Yes found it easy to position the cursor for text entry |
| Is the tab order on the form logical to the user | No, some areas not in the correct tab order and need re-arranging. |
| Can user save a specification form? | Yes no problems saving the file as a text file |
| Logging into the database? | Yes no problem with logging in |
| Can the user select each of the options on the welcome screen without any problems? | Liked the welcome screen with familiar and clear options available |
| Can you edit Supplier details? | Impressed with the instant saving of data |
| Can you delete a supplier? | Yes easy to use and liked how combo immediately refreshes the list of names so deleted supplier is not in the list |
| Can you edit factory details? | Yes, no problem |
| Can you search for a job and do the corresponding details of the job chosen show up across all forms? | Yes liked the search box that brings up the corresponding information. A bit confused by all the information being spread across so many forms but will get used to. |
| Can you edit a job? | Yes no problems faced and liked how leaving required fields empty produced error messages |

| TEST | FEEDBACK |
|---|---|
| Can a specification be loaded in? | The user was able to load in a specification form though did not know what was happening during the 15 second wait while the form was being loaded |
| Are you able to tell if the load has been successful? | Yes, scrolled through all the forms to check that the data was all there |
| Can you print a marketing report? | Happy with this feature as easy to use |
| Can you print a specification report for a particular job? | Impressed by clear layout of the document but would like to see it back on the original electronic specification form. |
| Is the on-line help useful? | Was very impressed to have this online. The document was found easy to look through with the hyperlinks provided on the contents page. |

# Chapter 8 - Deployment

The next stage in the process having completed testing is to install the system. The environment where the system is being deployed is Morrisons Head Office in Wakefield. Morrisons have agreed that the system should be installed on a trial basis to see how users respond to the system. If the trial period is successful, the system can be installed on a permanent basis or abandoned if not successful. The system comprises of the database and the electronic specification document. A demonstration will be given showing how to use the system and viewing the functionality of the system. The following sections discuss the details of implementing the database and electronic form.

## 8.1 The Database

The database will be installed on one personal computer (PC) at Morrisons Head Office where other databases are also stored. The database depending on the directory in which it will be stored will need to be configured accordingly. This involves changing path directories in macros implemented for the specification form load to the new path created. The VB code for deleting temporary files created because of the import feature also needs to be altered so that the directory path points to the correct directory. Data in the database used for testing needs to be deleted so that the system can be used. The database has been assigned a password that can be changed at the time of deployment to something more memorable for the users who own the system.

## 8.2 The Specification Form

The installation of the electronic specification form will be in a directory specified by Morrisons. Users will make a copy of the form every time it is sent out to a supplier with a hard copy remaining on file. A demonstration to be given to users will show how to save the document in a format for importing to the database and of the directory that the file must be saved in so that the database knows where to locate the file. Once the whole system has been deployed, there will be a cognitive walk-through of the system. This entails demonstrating all the functionality of the system and the tasks involved for each stage.

# Chapter 9 – Evaluation

## 9.1 Introduction

The evaluation of a system will vary depending on the objectives set out for the system for example in terms of cost or user satisfaction. The RAD methodology used to carry out this project emphasises the necessity for user involvement throughout the whole life cycle in order to produce a system that meets the user needs. This objective rather than cost has remained the focal point throughout the development process and consequently the evaluation criteria will be based around the success of the system in meeting user requirements. The evaluation of the system has been divided into two sections with the first section concentrating on the evaluation of the design carried out during the project and the second section looking at the evaluation of the final implementation. The reasoning behind this is to demonstrate how the results of design evaluation have been used to feed back into modifications to the design and hence try to improve the outcome of the implementation evaluation. The style of evaluation used for assessing both the design and implementation has been of laboratory studies. Inviting Duncan Roscoe from Morrisons to take part in an implementation evaluation session at the School of Computing has fulfilled the area of evaluation that requires user involvement.

## 9.2 Design Evaluation

Ideally, user evaluation should occur throughout the whole design process so that feedback can be incorporated into modifications of the design minimising the chances of having to make changes late on in the development process, which can be difficult. Due to time constraints discussed in the Project Progress section, it has not been possible to have direct user involvement for every part of the design of the system. Evaluation of the design tends to focus on using analytic techniques where the purpose is to identify areas that go against cognitive principles and overlook accepted empirical results such as HCI good design principles. The measures used to evaluate the design are discussed in the evaluation criteria section below.

### 9.2.1 Evaluation Criteria

The two approaches used for evaluating the design of the system are a cognitive walkthrough approach and heuristic evaluation techniques [36]. Both methods chosen are subjective as they rely upon the designer acting no behalf of the interests of the user than direct user participation. An attempt has been made to balance these methods and decrease bias using techniques that are more objective during the evaluation of the implementation. Cognitive walkthrough involves stepping through a series of actions on the interface for a task that a user needs to carry out. The purpose of this technique is to establish if the system is easy to learn by exploring around the system. The second

evaluation technique used is heuristic evaluation developed by Jakob Nielsen and Rolf Molich [37] whose design guidelines have been used for assessing the design of the system with the purpose of uncovering usability errors. The results of both evaluations are given below.

### 9.2.2 Results of evaluation

**Cognitive Walkthrough** – The tasks used for performing walkthroughs on the design of the access forms were taken from the system functionality section in Chapter 5, section. The only major usability problem uncovered in the earlier stages of design was that it was not obvious where on the interface a supplier could be edited or deleted. This was modified so that the action to edit or delete a supplier was placed on the main menu where the functionality could be easily seen. Using a cognitive walkthrough for evaluating the electronic specification form did not reveal problems as far as carrying out data entry on it was concerned.

**Heuristic evaluation** – The list of heuristics below has been used the design of the system with corrective measures detailed under each heading:

**1) Simple and natural dialog**

The design of the Access 2000 forms and electronic specification form all followed the natural order of the information as found on the paper-specification form.

**2) Speak the user language**

All system errors on the form have been converted into errors with messages containing terms that the user should be familiar with from their environment. This same principle is also used for help tips on the electronic specification form concerning data entry.

**3) Minimise user memory load**

**4) Be consistent**

The sequence required for moving between Access 2000 forms is the same on all of them

**5) Provide feedback**

When an action has been successful such as the import of a specification file in the database the user is taken to the form that reflects the load has been successful by showing the imported data

**6) Provide clearly marked exits**

All forms have exit buttons clearly positioned for exiting the system at any time they require.

**7) Provide short cuts**

**8) Good error messages**

All system error messages have been converted into user-friendly messages that return focus to the place where the error has occurred so that it can be fixed.

**9) Prevent errors**

System errors have been prevented by writing code that performs error checking. An action does not take place if a rule is violated and this stops an error message from appearing on the screen.

## 9.3 Implementation evaluation

The broader goals of evaluation [38] set out in relation to the user have been identified as:

**1) Assess the extent of system functionality**

**2) Assess the extent of the interface on the user**

**3) Identify any specific problems with the system**

These goals aim to provide objective methods of evaluation that can determine how successful the final system is. The criterion used for measuring the success of these goals is detailed under each goal heading in the following evaluation criteria section.

### 9.3.1 Evaluation Criteria

**1) Assess the extent of system functionality**

The evaluation of this goal involves seeing how well the objectives set out for the system have been achieved. This needs to be further expanded upon by evaluating how successful the solution is in order that objective testing is carried out. This is the aim of the evaluation criteria for goal two underneath.

**2) Assess the extent of the interface on the user**

The assessment carried out for this goal entails inspecting how easy the system is to learn, its usability and the user's perception of it. There are many concepts that can be used for measuring usability with the selected methods below identified as being amongst the most important:

- **Learnability** - How easy the user finds it to learn new tasks with only a user manual and minimal instructions
- **Familiarity** – How easy does the user find it to use the system in relation to carrying out similar tasks in the past
- **User Satisfaction** – Is the user satisfied with the system overall, the functionality it provides and the design.

**3) Identify any specific problems with the system**

Evaluation for achieving this goal is linked to the results of the tasks carried out for user acceptance in section 7.3.1. However, the focus has been to assess the negative aspects of the design that cause confusion to the user. The method used for this task has been observational involving user participation whereby seeking the user opinions and overall impression of the system.

### 9.3.2 Implementation Evaluation

**1) Assessing the extent of system functionality**

In order to assess the extent of system functionality and see how successful the solution is it is necessary to see if the objectives set out for the system have been reached. The objectives listed below confirm where requirements have been met along with further details where extra enhancements above the needs of the minimum requirements have been made.

- **Automate the data entry process from a set of paper forms to an electronic document.**

  This objective has been achieved with extra features implemented such as the "could have" requirement of password access for certain parts of the form filled in and extra features such as validation rules to reduce error in data entry implemented.

- **Design and implement a database to hold and view the data from specification forms.**

  An operational database has been implemented with an automatic loading mechanism for entering specification form data. A password has been assigned to the system as a security measure for restricting access to data. The implementation of forms in the database can be used to view the data held. A 'should have' requirement of reporting tools has also been implemented.

- **Ensure that the system is scalable to cope with an increasing amount of data and data held is maintainable.**

  The RDBMS used for this project has been evaluated in terms of its suitability for the number of records it is capable of holding. Access 2000 has been found to be over and above the needs required for this project and the user should not encounter any problems.

- **Provide a detailed user manual describing the operations of the system.**

  A user manual with screenshots detailing the functions of the system has been included in the Appendices. Access to this manual has been provided from within the database as a further enhancement.

**2) Assessing the extent of the interface on the user**

The usability measures were carried out after the user acceptance testing at a stage when it was thought that the user would be familiar with the system. The aim of this evaluation is to observe the user perform a series of tasks that can be measured against the usability principles identified.

Measure for learnability is **can the user learn tasks easily with only the user manual for instruction within 45 minutes?**

**Result:** The user was able to access the application and carry out most of the functionality. A process the user was not familiar with was how to fill in a Word Form but as time passed, more experience

was gained through exploration. Functionality of how to fill in the Word Form is not part of the user manual and should be incorporated in the future.

Measure for familiarity is **can the user identify between tasks once carried out on paper now computerised?**

**Result:** The user was not able to identify with the editing and deleting of suppliers, as this was not a process previously in place. Paper specification forms are filed by their job number and the user was able to use the search tool and select a job number to view the specification.

Measure for the user satisfaction is the following questions based on a scale of 1-5 (1 good, 5 bad). The response must be between 1-3 to be measured as successful.

1) **Are you satisfied with the system?**
2) **Do you find the system easy to use? Is it intuitive enough?**
3) **How useful do you find the user manual?**
4) **Would you use it in the future?**

Results:

1) The user was satisfied overall, but would like to see more work done on the reporting tools. Scored 3
2) The system itself was found easy to use but the user will require more guidance on how to use the electronic form and the various features. Scored 2
3) The user found the manual useful for the tasks involving the database but it did not cover areas on using the electronic form. Scored 4
4) The user would use the system in the future but alongside the old paper process until further evidence of the system reliability is seen. Scored 2

**3) Identifying any specific problems with the system**

The method involved in achieving this goal was to observe the user using the system as part of the user acceptance testing. Any significant problems identified because of the testing discussed in section 7.3.1 have been noted below in order to incorporate as part of further software releases in the future.

## 9.3 Overall Summary

The evaluation of the system has not produced any unexpected results due to the user feedback gained during the implementation that helped to ensure that the project is developed in the right direction as required by the user. The user on overall was impressed with the system though there are a few areas of the system that need improvement for complete user satisfaction. A further evaluation of the system will need to be undertaken once the system is installed in order to conclusively prove if the system is a success or not.

# Bibliography

[1] Cornford, Tony, Avergerou, Chrisanthi, (1998), **Developing Information Systems; Concepts, Issues and Practice,** Macmillan.

[2] Connell'O, Fergus, (1996), **How to Run Successful Projects**, pp.1-3, pp.73-78, pp.89-99.

[3] Lau, Henry, (2001), **Analysis and design of an information system for AIS,** University of Leeds, School of Computer Studies

[4] Pearse, Adam, (2001), **An automated sports centre booking system,** University of Leeds, School of Computer Studies

[5] Mott, Peter, Roberts, Stuart, (1998), **DB11 Introduction to Databases. Module Notes,** School of Computing, University of Leeds.

[6] Dix, Alan, Finlay, Janet, Abowd, Gregory, Beale, Russell, (1998), **Introduction to Human Computer Interaction, 2nd Edition**, Prentice Hall, pp.143-162, pp.179-209.

[7] Avison and Fitzgerald, (1995), **Information Systems Development: Methodologies, Techniques and Tools**, Mc-Graw Hill Companies, pp.10.

[8] Avison and Fitzgerald, (1995), **Information Systems Development: Methodologies, Techniques and Tools**, Mc-Graw Hill Companies, pp.11-12.

[9] Avison and Fitzgerald, (1995), **Information Systems Development: Methodologies, Techniques and Tools**, Mc-Graw Hill Companies, pp.294-303.

[10] Avison and Fitzgerald, (1995), **Information Systems Development: Methodologies, Techniques and Tools**, Mc-Graw Hill Companies, pp.391-399.

[11] Dix, Alan, Finlay, Janet, Abowd, Gregory, Beale, Russell, (1998), **Introduction to Human Computer Interaction, 2nd Edition**, Prentice Hall, pp.205.

[12] Dix, Alan, Finlay, Janet, Abowd, Gregory, Beale, Russell, (1998), **Introduction to Human Computer Interaction, 2nd Edition**, Prentice Hall, pp.206-207.

[13] Stapleton, J, (1999), **Dynamic Systems Development Method**, Addison-Wesley, pp.23

[14] Stapleton, J, (1999), **Dynamic Systems Development Method**, Addison-Wesley, pp.3-10.

[15] Stapleton, J, (1999), **Dynamic Systems Development Method**, Addison-Wesley, pp.28-29.

[16] Avison, D, E, (1992), **Information Systems Development**, Blackwell Scientific Publications, pp.77.

[17] Avison, D, E, (1992), **Information Systems Development**, Blackwell Scientific Publications, pp.78-79.

[18] Avison, D, E, (1992), **Information Systems Development**, Blackwell Scientific Publications, pp.91.

[19] Avison, D, E, (1992), **Information Systems Development**, Blackwell Scientific Publications, pp.102.

[20] Avison, D, E, (1992), **Information Systems Development**, Blackwell Scientific Publications, pp.104.

[21] Atzeni, P, Ceri, S, Paraboschi, S, Torlone, R, (1999), **Database Systems, Concepts, Languages and Architectures**, Mc-Graw Hill Companies.pp.320-349.

[22] Avison, D, E, (1992), **Information Systems Development**, Blackwell Scientific Publications, pp108.

[23] Dix, Alan, Finlay, Janet, Abowd, Gregory, Beale, Russell, (1998), **Introduction to Human Computer Interaction, 2nd Edition**, Prentice Hall, pp.143-162, pp.406-408.

[24] Elmasri and Navathe, (2000), **The Fundamentals of Database Concepts,** Addison-Wesley**,** pp.545.

[25] Elmasri and Navathe, (2000), **The Fundamentals of Database Concepts,** Addison-Wesley**,** pp.546.

[26] Faulkner. C., (1998), **The Essence of Human Computer Interaction**, Prentice Hall, pp.22.

[27] Faulkner. C., (1998), **The Essence of Human Computer Interaction**, Prentice Hall, pp.19-28.

[28] Weiss, Edmond, (1985), **How to write a Usable User Manual**, ISI Press, pp.32-33.

[29] Elmasri and Navathe, (2000), **The Fundamentals of Database Concepts,** Addison-Wesley**,** pp.449.

[30] Avison, D, E, (1992), **Information Systems Development**, Blackwell Scientific Publications, pp.115-122.

[31] Norton, Peter, Andersen, V, (2000**)**, **Guide to Microsoft Access 2000 Programming**, Techmedia, pp.307-342.

[32] Dix, Alan, Finlay, Janet, Abowd, Gregory, Beale, Russell, (1998), **Introduction to Human Computer Interaction, 2nd Edition**, Prentice Hall, pp.449.

[33] Kit,Edward, (1995), **Software Testing in the Real World**, Addison-Wesley, pp.77-109.

[34] Pfleeger, Lawrence, Shari, **Software Engineering: The Product of Quality Software**, Macmillan Publishing, (1987), pp.265-293, 311-332.

[35] Pfleeger, Lawrence, Shari, **Software Engineering: The Product of Quality Software**, Macmillan Publishing, (1987), pp.330-335.

[36] Dix, Alan, Finlay, Janet, Abowd, Gregory, Beale, Russell, (1998), **Introduction to Human Computer Interaction, 2nd Edition**, Prentice Hall, pp.409-414.

[37] Dix, Alan, Finlay, Janet, Abowd, Gregory, Beale, Russell, (1998), **Introduction to Human Computer Interaction, 2nd Edition**, Prentice Hall, pp.413-415.

[38] Dix, Alan, Finlay, Janet, Abowd, Gregory, Beale, Russell, (1998), **Introduction to Human Computer Interaction, 2nd Edition**, Prentice Hall, pp.406.

# Appendix A

Overall, I found the project experience rewarding and challenging though I was disappointed about the amount of project slippage. Using a methodology was helpful as it provided a strong structure for the project and gave me a sense of direction so that I felt confident about the project approach. The RAD methodology itself was very effective until the point of slippage but after that point it was very hard to find time to incorporate user feedback due to the short amount of time left to implement the system. Consequently, this led to user expectations being higher at the time of testing, as there may have been the feeling that the system had been developed in secrecy. Given the opportunity, again I would ensure that more time was spent on the analysis and design with user participation than the implementation stage. I felt that these two sections needed to be more thorough as it would have led to encountering fewer problems during the implementation. To students doing similar projects in the future I would offer the following advice:

Do not let lack of knowledge in programming or with the software put you off a project. The unique factor about this project is that I was using common software packages to perform less conventional tasks than what a normal user would use this software for on a day-to-day basis. This included exploring the kind of functionality that most help books did not think to write about. Although the functionality required for the project was not difficult in theory, in practice it was lot more complicated and quite thought provoking to find ways to achieve certain tasks such as the load mechanism.

Try not to rush through the analysis as it can save a lot of misunderstanding later on in the process when it is more difficult to incorporate changes. It was not until the design phase through user feedback on the design of the forms that I found out about certain information that should have been part of the analysis process.

Plan the time you intend to spend on the key tasks and keep to it. As I had a project plan, when I did suffer project slippage I was able to adjust my tasks accordingly on the Gantt chart and plan ahead instead of losing focus on the remaining tasks.

User participation is vital to the success of the system therefore try to include them as much as you can during the process. That way you are ensuring that what you implement is actually what they want. I found that as time passed without user involvement, expectations of the final system grew higher.

Lastly, choose a project that genuinely interests you. This is probably the most important requirement, as when you come across difficulties or suffer setbacks it is your enthusiasm and desire to solve the problem that helps you persevere and carry on. I found the project very interesting and this helped me to understand and overcome the problems I faced during the Implementation of the database.

# Appendix B Meeting Details

Overview of meeting with Duncan Roscoe from Morrisons Head Office
Date: Tuesday 30th October 2001
Duration: 2 hours
Attendees: Leena Patel, Duncan Roscoe, and Stuart Roberts

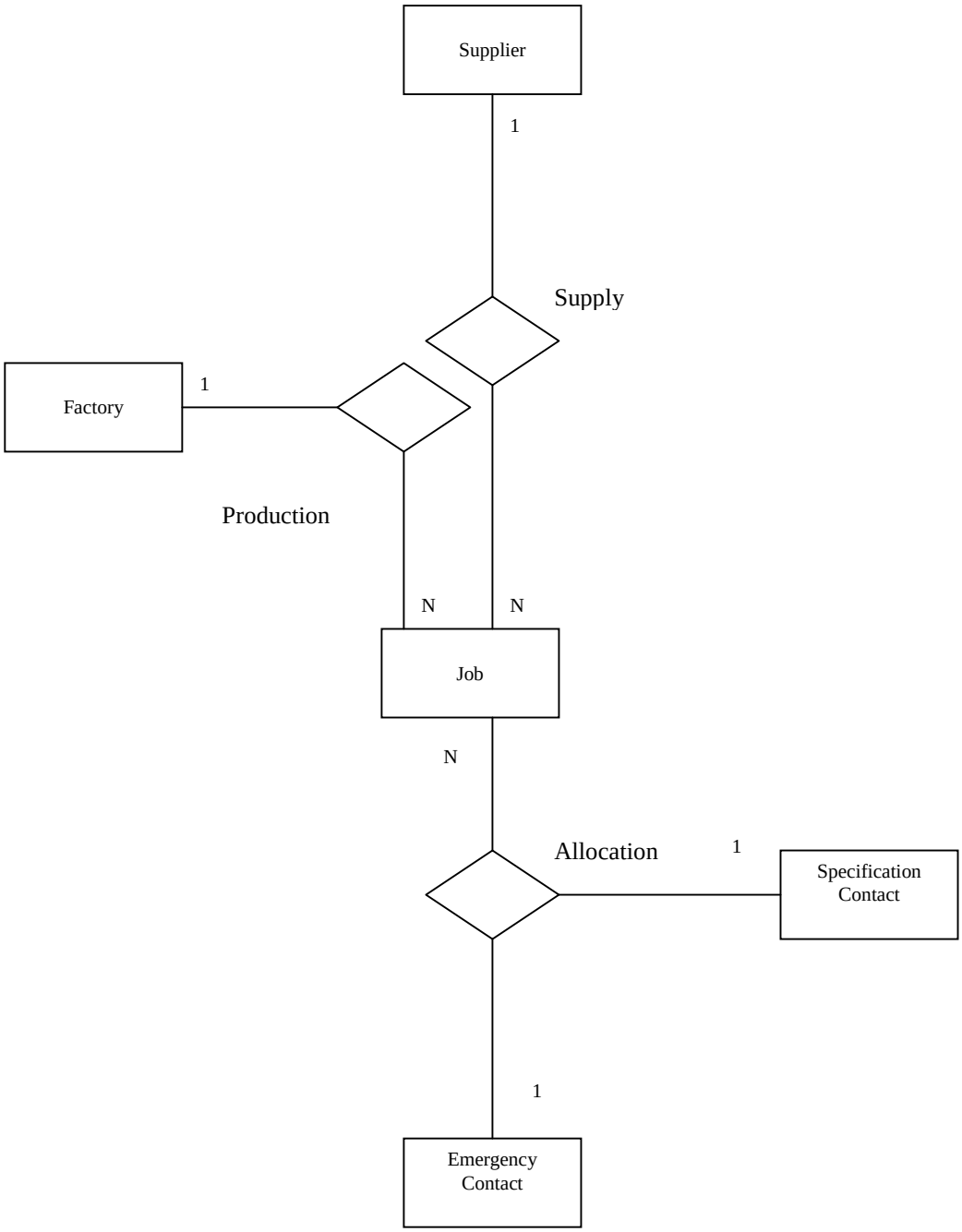
**1) What is the current process in place?**
A team of buyers and suppliers agree on the decision to sell a new branded products and certain information is pre-agreed such as the claims information and Microbiological Specification for the product. This is information is filled in on a paper Specification form, assigned a unique job number and then sent out to a supplier who is required to fill in the rest of the form. The problem faced is that some suppliers change some of the pre-filled in information and this is not allowed. On return of the form, the form is checked over before it is either approved or abandoned and then filed away in a filing cabinet.

**2) What would be your requirements for an automated system?**
We would require that the specification form be provided electronically to email to suppliers. Any system implemented cannot grant network access to the suppliers due to security issues. Another requirement is that at present, it is not possible to search for information and therefore we would like to be able to search for information and create ad-hoc reports that can be emailed to other departments. It would also be useful to have a feature where the user cannot change information that has already been filled in on a specification form. If it is possible we would like to track deleted specification forms and highlight certain answers for fields if it dies not match our expectations.

**3) Where will this system be implemented and how many potential users of the system?**
The system should be implemented at the head office in Wakefield and would have around 3-7 users. We would expect the users to be granted password access so that there is no authorised access to the database.

**4) Do you have any software requirements for the system?**
We prefer that Microsoft Word be used for the forms as this is the software that we expect most of our suppliers have and different versions tend to support all documents. Access is the current software that holds all of our other databases and from a linking point of view, in the future this would be ideal. We are prepared to but other software if needed within a reasonable amount of money.

# Appendix C ER-Diagram



# Appendix D ER-Modelling Documentation

| ENTITY | DESCRIPTION | ATTRIBUTE | KEY |
|---|---|---|---|
| Factory | Independent factory chosen to produce a job | Factory name<br>Address<br>Telephone | Factory name |
| Job | Job on which Supplier and Factory are working | Job no<br>Product_name<br>Food name<br>Supplier comments<br>Protective_atmosphere<br>Life of product once open<br>Health mark<br>Shelf life<br>Shelf life format<br>Sequence of packaging<br>Serving instructions<br>Primary_trans_pack_inlimit<br>Return date<br>Approved date<br>Initials<br>HACCP study<br>Shelf trials<br>Weight<br>Drained weight<br>Volume<br>Count<br>Not applicable<br>E mark<br>Qty approx<br>Irradiated<br>GM<br>Suitable for veg<br>Doc procedure for package<br>Spec complete date<br>RDA Field<br>NutInfo<br>PerServing | Job no |
| Supplier | Independent supplier chosen to supply a job | Supplier_no<br>Supplier name<br>Address<br>Telephone | Supplier_no |
| Specification Contact | Contact assigned to a job who works for the supplier or factory | Spec Name<br>Job Title<br>Job_no<br>Tel<br>Email | Spec Name |
| Emergency Contact | Emergency contact assigned to a job<br>Who works for the supplier or factory | Emerg name<br>Job title<br>Job_no<br>Tel | Emerg name |

| ENTITY | DESCRIPTION | ATTRIBUTES | KEY |
|---|---|---|---|
| Product Claims info | Describes the product | Claims_info | |
| Product Ingredient | Describes the ingredients in the product | Ingredient<br>Quid | |
| Warnings Table | Describes warning label of product | warnings | |
| Product Composition | Describes the composition of the product | Analysis<br>Frequency<br>Reason | |
| Product Nutrition | Describes the nutrition value of product | Category<br>Qty per grams<br>Qty per servings<br>energy kjperg<br>energy kcalperg<br>energy kjperserv<br>energy kjcalperserv | |
| Product Formulation | Describes the formulation of ingredients in the product | Ingredient formulation<br>Ingredient description<br>Country of origin<br>percentage | |
| Product allergens | Describes the allergens in the product | Allergens<br>Contains or not | |
| Product Microbiological Specifications | Describes the required specification of the product | **Mspec** | |

| RELATIONSHIP | DESCRIPTION | ENTITIES INVOLVED | ATTRIBUTES |
|---|---|---|---|
| SUPPLIES | Assigns a supplier to a job | Supplier(1,N)<br>Job( 0,1) | |
| PRODUCES | Associates a factory with a job | Factory(1,N)<br>Job (0,1) | |
| ALLOCATES | Allocates a Specification Contact and 24H Emergency Contact to a Job | Job (1,1)<br>Specifcation Contact(1, N)<br>Emergency Contacy(1,N) | |
| CONTAINS | Every job contains<br>Claims info<br>Ingredients<br>Warnings<br>Composition info<br>Nutrition value<br>Formulation<br>Microbiological Specifications<br>Allergens | Job(1,N)<br>Claims info (N,1)<br>Ingredients (N,1)<br>Warnings (N,1)<br>Composition info (N,1)<br>Nutrition value (N,1)<br>Formulation (N,1)<br>Microbiological Specifications (N,1)<br>Allergens (N,1) | |

# Appendix E Database Schema

**JOB**

| ATTRIBUTE | ATTRIBUTE TYPE | DESCRIPTION |
|---|---|---|
| Job no | Text | Primary key for job table |
| Supplier_no | Text | Foreign key to Supplier table |
| Factory_name | Text | Foreign key to Factory table |
| Product_name | Text | Name of product |
| Food name | Text | True name of product |
| Supplier comments | Memo | Comments |
| Protective_atmosphere | Text | Production requirements |
| Life of product once open | Text | Describes the product |
| Health mark | Text | Describes the product |
| Shelf life | Text | Describes the product |
| Shelf life format | Text | Describes the product |
| Sequence of packaging | Memo | Packaging requirements |
| Serving instructions | Memo | How the product is served |
| Primary_trans_pack_inlimit | Text | Packaging requirements |
| Material reuse | Text | Packaging requirements |
| Noxious_sub_ltd | Text | Packaging requirements |
| Mats_comply | Text | Packaging requirements |
| Plastic_food comply | Text | Packaging requirements |
| Product group | Text | Describes the product |
| Trading mgr | Text | Manager in charge of Job |
| Issue date | Text | Date specification issued |
| Return date | Text | Date form returned by supplier |
| Approved date | Text | Date of go-ahead by Morrisons |
| Initials | Text | Sign-off from Manager |
| HACCP study | Text | Production requirements |
| Shelf trials | Text | Production requirements |
| Weight | Text | Describes the product |
| Drained weight | Text | Describes the product |
| Volume | Text | Describes the product |
| Count | Text | Describes the product |
| Not applicable | Text | Describes the product |
| E mark | Text | Production requirements |
| Qty approx | Text | Production requirements |
| Irradiated | Text | Production requirements |
| GM | Text | Describes the product |
| Suitable for veg | Text | Describes the product |
| Doc procedure for package | Text | Production requirements |
| Spec complete date | Text | Date specification complete |
| RDA Field | Memo | Describes the product |
| NutInfo | Text | Grammage of product |
| PerServing | Text | Energy value of product |

**SUPPLIER**

| ATTRIBUTE | ATTRIBUTE TYPE | DESCRIPTION |
|---|---|---|
| Supplier_number | Text | Primary key for Supplier table |
| Supplier_name | Text | Name of Supplier |
| Address | Text | Address of Supplier |
| Tel | Text | Telephone number of Supplier |

**CLAIMS TABLE**

| ATTRIBUTE | ATTRIBUTE TYPE | DESCRIPTION |
|---|---|---|
| Job_no | Text | Foreign Key for Job table |
| Claims_info | Text | Describes the product |

**FACTORY**

| ATTRIBUTE | ATTRIBUTE TYPE | DESCRIPTION |
|---|---|---|
| Factory_name | Text | Primary Key for Factory table |
| Address | Text | Address of Factory |
| Tel | Text | Telephone of Factory |

**24H EMERGENCY CONTACT**

| ATTRIBUTE | ATTRIBUTE TYPE | DESCRIPTION |
|---|---|---|
| Emerg_ Name | Text | Name of contact assigned to job |
| Job Title | Text | Role of the contact |
| Job_no | Text | Foreign key for Job title |
| Tel | Text | Direct telephone line of contact |

**SPECIFICATION CONTACT**

| ATTRIBUTE | ATTRIBUTE TYPE | DESCRIPTION |
|---|---|---|
| Spec Name | Text | Name of contact assigned to job |
| Job Title | Text | Role of the contact |
| Job_no | Text | Foreign key for the job table |
| Tel | Text | Direct telephone line of contact |
| Email | Text | Email details of contact |

**PRODUCT INGREDIENT**

| ATTRIBUTE | ATTRIBUTE TYPE | DESCRIPTION |
|---|---|---|
| Job_no | Text | Foreign key for the job table |
| Ingredient | Text | Ingredients in the product |
| Quid | Text | Percentage of ingredients |

**WARNINGS**

| ATTRIBUTE | ATTRIBUTE TYPE | DESCRIPTION |
|---|---|---|
| Job_no | Text | Foreign key for the job table |
| Warnings | Memo | Product food warnings |

**PRODUCT MICROBIOLOGICAL SPECIFICATIONS**

| ATTRIBUTE | ATTRIBUTE TYPE | DESCRIPTION |
|---|---|---|
| Job_no | Text | Foreign key for the job table |
| Mspec | Memo | Set specification details for job |

**PRODUCT FORMULATION**

| ATTRIBUTE | ATTRIBUTE TYPE | DESCRIPTION |
|---|---|---|
| Job_no | Text | Foreign key for the job table |
| Ingredient formulation | Text | Ingredient name |
| Ingredient description | Text | Description of ingredient |
| Country of origin | Text | Describes origin of ingredient |
| Percentage | Text | Percentage of ingredients |

**PRODUCT COMPOSITION**

| ATTRIBUTE | ATTRIBUTE TYPE | DESCRIPTION |
|---|---|---|
| Job_no | Text | Foreign key for Job table |
| Analysis | Text | Describes the product |
| Frequency | Text | Tests carried out on product |
| Reason | Memo | Reasons for the test |

**PRODUCT NUTRITION**

| ATTRIBUTE | ATTRIBUTE TYPE | DESCRIPTION |
|---|---|---|
| Job_no | Text | Foreign key for the job table |
| category | Text | List of nutrition values |
| qty per grams | Text | Amount of nutrition in product |
| qty per servings | Text | Amount of nutrition per serving |
| energy kjperg | Text | Amount of energy(kj) in product |
| energy kcalperg | Text | Value of energy(kcal) in product |
| energy kjperserv | Text | Amount of energy(kj) in product |
| energykcalperserv | Text | Value of energy(kcal) in product |

**PRODUCT ALLERGENS**

| ATTRIBUTE | ATTRIBUTE TYPE | DESCRIPTION |
|---|---|---|
| Job_no | Text | Foreign key for the Job table |
| Allergens | Text | Allergen checklist |
| Contains_or_not | Text | YES/NO to allergen checks |

# Appendix E Testing Results

## Results of Unit Testing

### WELCOME SCREEN

| TEST | RESULT |
|---|---|
| Does load specification form command button direct user to the right form? | Yes, directed to the first form of Job specification where current job details shown are of those loaded in by previous action. |
| Does edit/delete supplier command button direct the user to the supplier form? | Yes, directed to the supplier form where supplier details shown are those of first supplier record in the database |
| Does edit/delete factory command button direct the user to the factory form? | Yes, directed to factory form where supplier details shown are those of first factory record in the database. |
| Does the print marketing report command button bring up a pop-up screen asking the job number to print a report on? | Yes, a pop-screen appears where the user is required to type in the job number they want to base a report on. |
| If a job number is entered that is not in the database, does an error come up? | No, a blank report is produced. |
| Does the print specification report command button bring up a pop-up screen asking the job number to print a report on? | Yes, a pop-screen appears where the user is required to type in the job number they want to base a report on. |
| If a job number is entered that is not in the database, does an error come up? | No, a blank report is produced. |
| On opening the welcome screen is maximised but can it be minimised and does the format remain unaffected with all features still in view | Yes, all the features remain in view. |

### JOB FORM

| TEST | RESULT |
|---|---|
| Are all correct details on view? | Yes, all correctly shown from database |
| Can a job be selected to view details on other than the one that shows when the form is opened? | Yes, job can be selected using Combo Box and form is updated to reflect job number chosen |
| Does Combo Box show the correct job numbers in the database? | Yes, all job numbers in the database are shown in the list |
| What happens if a job number not in the database is typed into the Combo Box? | Job number typed in remains in display but nothing happens to form. When user pulls list of Combo Box down the wrong number vanishes. |
| If the delete of job is successful, does it show on the screen? | No, but the job number deleted is no longer in the Combo Box so that it cannot be selected. |
| Is error message generated if key fields left empty. | Yes, error message generated for each field violated by null value. |

**24H EMERGENCY CONTACT FORM**

| TEST | RESULT |
|---|---|
| Are correct values, according to job number shown on job specification form? | Yes, all values shown correctly from database |

**SPECIFICATION FORM**

| TEST | RESULT |
|---|---|
| Are correct values, according to job number shown on job specification form? | Yes, all values shown correctly from database |

**SUPPLIER FORM**

| TEST | RESULT |
|---|---|
| Are correct values, according to job number shown on job specification form? | Yes, all values shown correctly from database |

**FACTORY FORM**

| TEST | RESULT |
|---|---|
| Are correct values, according to job number shown on job specification form? | Yes, all values shown correctly from database |

**SUPPLIER COMMENTS FORM**

| TEST | RESULT |
|---|---|
| Are correct values, according to job number shown on job specification form? | Yes, all values shown correctly from database |

**CLAIMS FORM**

| TEST | RESULT |
|---|---|
| Are correct values, according to job number shown on job specification form? | Yes, all values shown correctly from database |

**NUTRITION FORM**

| TEST | RESULT |
|---|---|
| Are correct values, according to job number shown on job specification form? | Yes, all values shown correctly from database |

**RDA FORM**

| TEST | RESULT |
|---|---|
| Are correct values, according to job number shown on job specification form? | Yes, all values shown correctly from database |

**INGRIEDIENTS FORM**

| TEST | RESULT |
|---|---|
| Are correct values, according to job number shown on job specification form? | Yes, all values shown correctly from database |

**PRODUCT INFORMATION FORM**

| TEST | RESULT |
|---|---|
| Are correct values, according to job number shown on job specification form? | Yes, all values shown correctly from database |

**COMPOSITION FORM**

| TEST | RESULT |
|---|---|
| Are correct values, according to job number shown on job specification form? | Yes, all values shown correctly from database |

**PREPARATION AND COOKING FORM**

| TEST | RESULT |
|---|---|
| Are correct values, according to job number shown on job specification form? | Yes, all values shown correctly from database |

**FORMULATION FORM**

| TEST | RESULT |
|---|---|
| Are correct values, according to job number shown on job specification form? | Yes, all values shown correctly from database |

**PACKAGING FORM**

| TEST | RESULT |
|---|---|
| Are correct values, according to job number shown on job specification form? | Yes, all values shown correctly from database |

**EDIT/DELETE SUPPLIER**

| TEST | RESULT |
|---|---|
| Are all the correct suppliers shown in Combo Box? | Yes, all values shown from database |
| When supplier is deleted, does it show deletion is successful? | No, but Combo Box is refreshed so that it does not display deleted supplier in the list |
| When supplier is edited, does it show details have been saved? | No, but the edited details remain on screen and look saved. |

**EDIT/DELETE FACTORY**

| TEST | RESULT |
|---|---|
| Are all the correct factories shown in Combo Box? | Yes, all values shown from database |
| When a factory is deleted, does it show deletion is successful? | No, but Combo Box is refreshed so that it does not display deleted factory in the list |
| When factory is edited, does it show details have been saved? | No, but the edited details remain on screen and look saved. |

**ELECTRONIC SPECIFICATION FORM**

| TEST | RESULT |
|---|---|
| If the user enters wrong format in data entry fields with validation rules is an error message generated. | Yes a message is generated instructing the value or format that data needs to be in. |
| When a filled in form is saved in text only format is all the data retained? | Yes, all the data from the form is present in the text file. |
| Do data entry fields show help tips when typing in text boxes? | Yes, messages instructing the type of data entry expected in the text box are displayed in the left hand corner of the screen. |

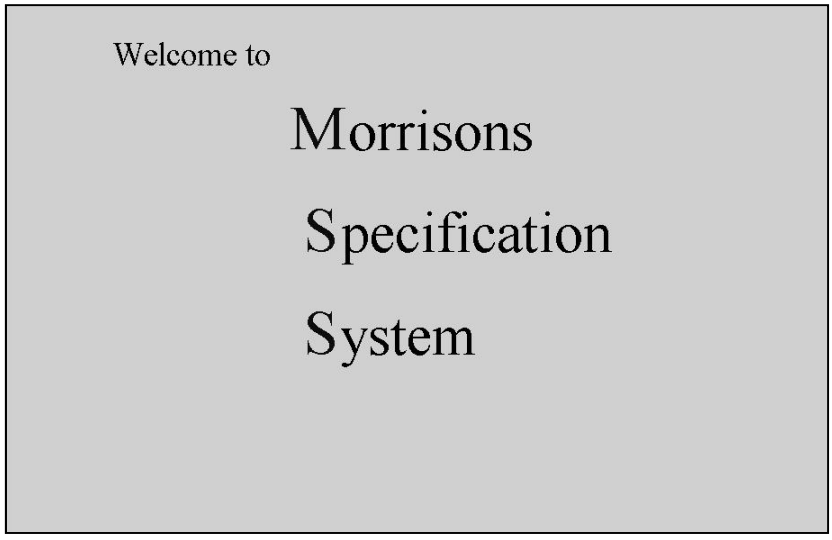
## Results of Integration Testing

**LOAD OF SPECIFICATION FORM**

| TEST | RESULTS |
|---|---|
| When loading specification form in are the following events successful<br>1. Splitting of text file into smaller text file<br>2. Import procedure using macros<br>3. Append of data from temporary tables to correct tables<br>4. Deletion of temporary tables<br>5. Deletion of text files created<br>6. Correct information shown by form from import | 1. Yes and no data has been lost<br>2. All data imported without any error message<br>3. Append is successful however occasionally error messages have appeared that state the append has not been successful. Despite the error on further investigation all data has still been appended with no obvious errors.<br>4. Yes all temporary tables get deleted<br>5. Yes all text files are deleted<br>6. The form reflect all the correct data held in database |
| Does an error message come up if a text file cannot be found in the location specified in the macro for either splitting or deleting text files? | Yes, an error message is generated that states the file has not been found in the directory path expected. The full path is also shown so that the user can move the file to the correct directory to run the procedure again |

# Appendix G User Manual

## USER MANUAL FOR MORRISONS SPECIFICATION SYSTEM



## CONTENTS

### Introduction:

This user manual has been designed for use in conjunction with Morrisons Specification System. The guide aims to introduce the system by helping the user perform key functions. Click on the hyperlink below to see how to do a task.

1) How to Load A Specification Form into the database
2) How to search for a Specification Form by the Job Number
3) How to Edit/Delete a supplier
4) How to Edit/Delete a factory
5) How to print a Marketing report
6) How to Print a Job Specification report

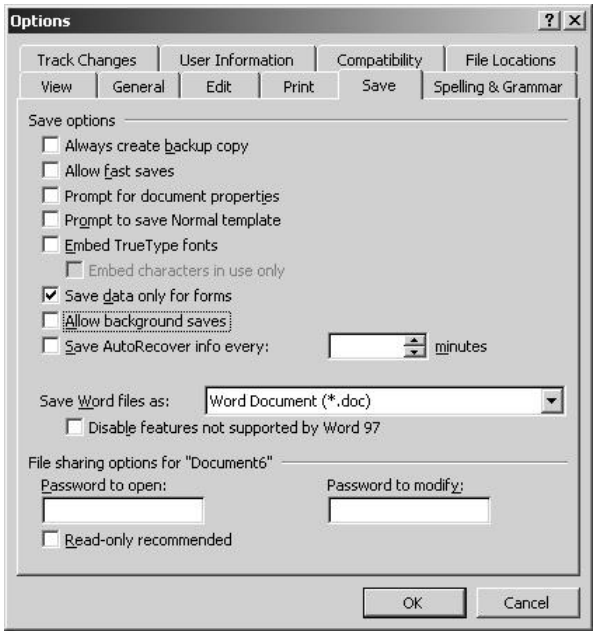
## 1) HOW TO LOAD A SPECIFICATION FORM INTO THE DATABASE

**To load a form into the database, the specification form must be prepared for the load. How do I do this? (Follow the steps below)**

a) Open the Specification Form in Microsoft Word that you want to load into the database.

b) Go to the Tools menu at the top of the screen, expand the menu and select 'Options'

c) Click on the save tab and ensure that the check box 'Save data for forms' is checked and then press 'OK' to confirm as shown in the diagram below




d) Then click on the File menu and select 'Save As'

e) Give the file a name and save it in the Morrisons Specifications folder as a 'text only' file

**The file is now ready to import into the database!**

1) Open the Morrisons Specification System database

2) Enter the password to gain access to the system

3) On the Welcome Screen click on the 'Load Specification Form' option and wait to be directed to the job form that shows the load has been successful.

## TROUBLESHOOTING

1) **'File not found error'**

**Check that the text file is in the directory that it should be.**

## 2) HOW TO SEARCH FOR A SPECIFICATION FORM

1) On the welcome screen of the application, click on the 'Search for a Job Specification', option.
2) On the Job form select a job number from the pull-down list to see the associated details for that Specification

## 3) HOW TO EDIT/DELETE A SUPPLIER

**How do I do this? (Follow the steps below)**

On the welcome screen of the application click on the 'Edit/Delete Supplier', option.

**To edit a supplier do:**

A) Select a Supplier name from the pull down list, make any changes and exit the form. All changes are saved on exit.

**To delete a Supplier do:**

B) Select a Supplier name from the pull down list and press the delete button to delete the Supplier from the database.

## 4) HOW TO EDIT/DELETE A FACTORY

**How do I do this? (Follow the steps below)**

1) On the welcome screen of the application click on the 'Edit/Delete Factory' option

**To edit a factory:**

A) Select a factory name from the pull down list, make any changes and exit the form. All changes are saved on exit

**To delete a factory:**

B) Select a factory name from the pull down list and press the delete button to delete Factory from the database

## 5) HOW TO PRINT A MARKETING REPORT

**How do I do this? (Follow the steps below)**

1) On the welcome screen select the 'Print a Marketing Report' option.
2) Enter a Job number in the pop-up box and press 'OK' to create a report

## TROUBLESHOOTING

1) A blank report is created
   Check, that the job number entered to create a report is stored in the database

## 6) HOW TO PRINT A SPECIFICATION REPORT

**How do I do this? (Follow the steps below)**

1) On the welcome screen select the ‘Print a Job Specification Report’
2) Enter a Job number in the pop-up box and press ‘OK’ to create a report

## TROUBLESHOOTING

1) A blank report is created
   Check that the job number entered to create a report is in the stored in the database